



US006697865B1

(12) **United States Patent**
Howard et al.

(10) Patent N .: **US 6,697,865 B1**
(45) Date of Patent: **Feb. 24, 2004**

(54) **MANAGING RELATIONSHIPS OF PARTIES INTERACTING ON A NETWORK**

(75) Inventors: **Garry W. Howard**, Cambridge, MA (US); **Michael R. Gretzinger**, Jamaica Plain, MA (US)

(73) Assignee: **E.piphany, Inc.**, San Mateo, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/477,254**

(22) Filed: **Jan. 4, 2000**

(51) Int. Cl.⁷ **G06F 15/16**

(52) U.S. Cl. **709/229; 709/203; 709/217; 709/228; 707/9; 713/200**

(58) Field of Search **709/200-203, 709/217-219, 227-229; 713/201-202, 152, 155, 166; 707/1, 9-10**

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,678,041 A * 10/1997 **Baker et al.** **709/229**

5,815,665 A * 9/1998 **Teper et al.** **709/229**
5,822,518 A * 10/1998 **Ooki et al.** **709/229**
5,826,268 A * 10/1998 **Schaefer et al.** **707/9**
6,092,196 A * 7/2000 **Reiche** **709/229**
6,105,063 A * 8/2000 **Hayes, Jr.** **709/228**
6,189,104 B1 * 2/2001 **Leppek** **709/229**
6,339,826 B2 * 1/2002 **Hayes, Jr. et al.** **713/166**

* cited by examiner

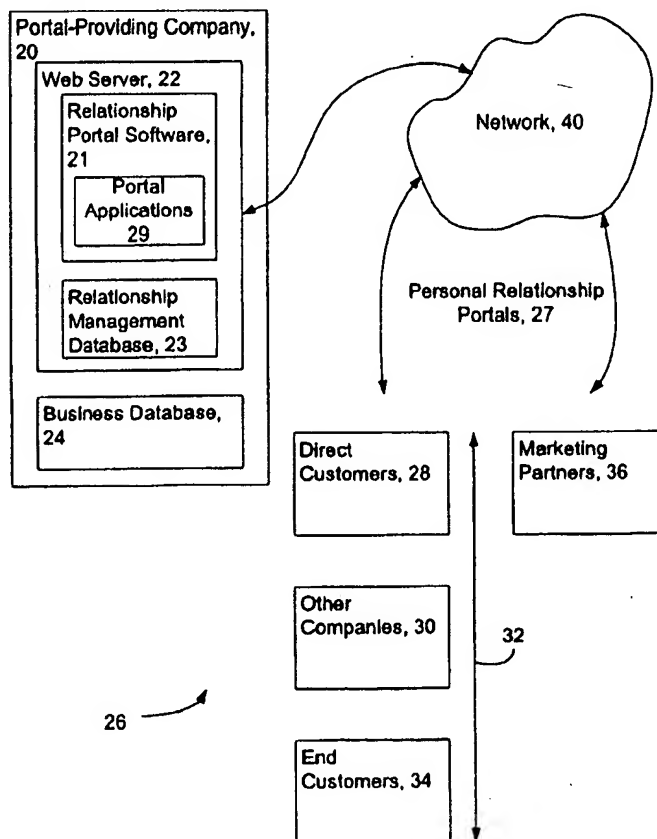
Primary Examiner—Bharat Barot

(74) **Attorney, Agent, or Firm—Bingham McCutchen LLP; Fabio E. Marino; Jeffrey S. Smith**

(57) **ABSTRACT**

Information is stored that reflects the existence of relationships between identified parties with respect to use of the digital facility. A predetermined type of interaction of the parties is permitted) via the electronic communication medium with respect to the digital facility if the stored information reflects the existence of relationships between them.

27 Claims, 3 Drawing Sheets



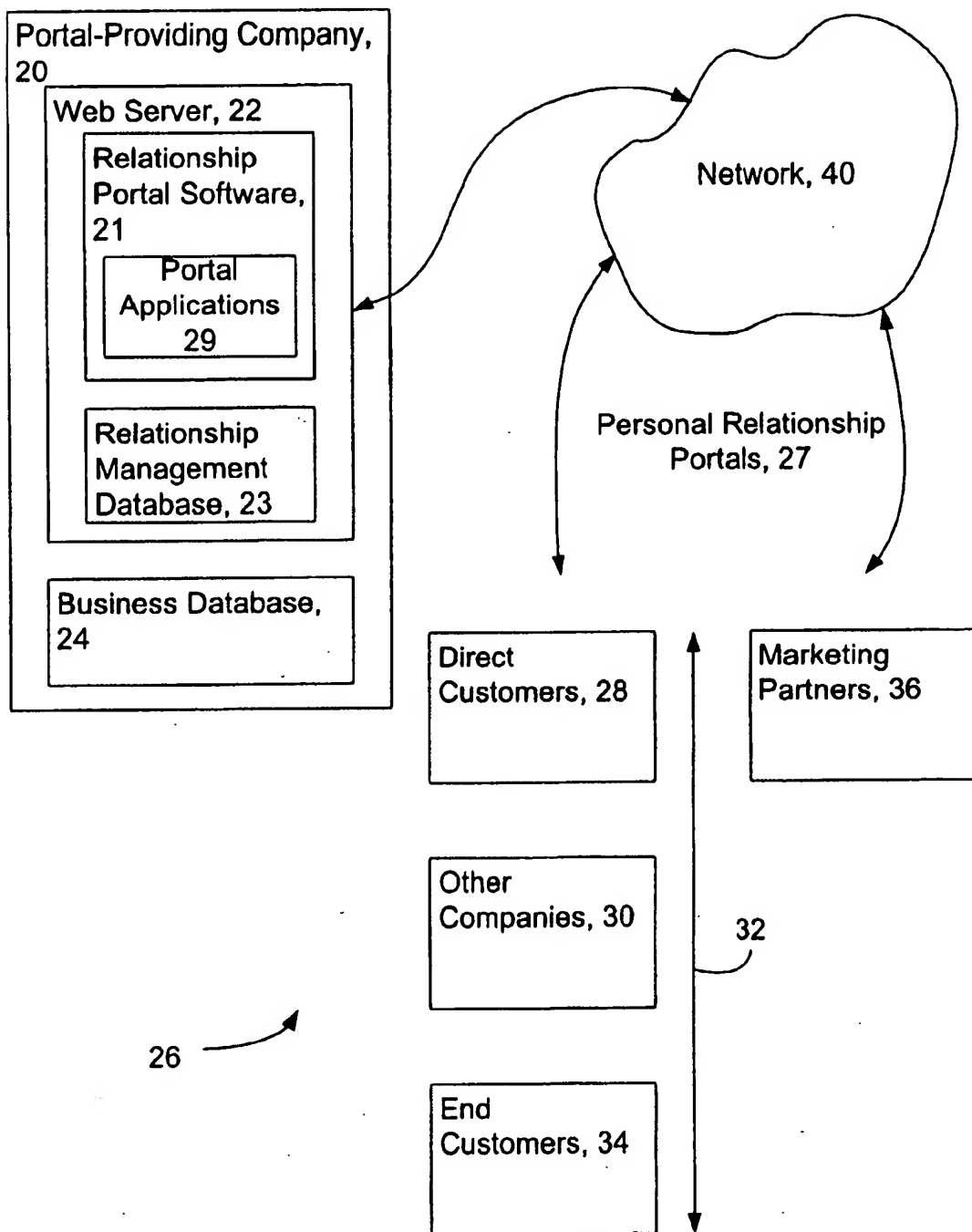


Figure 1

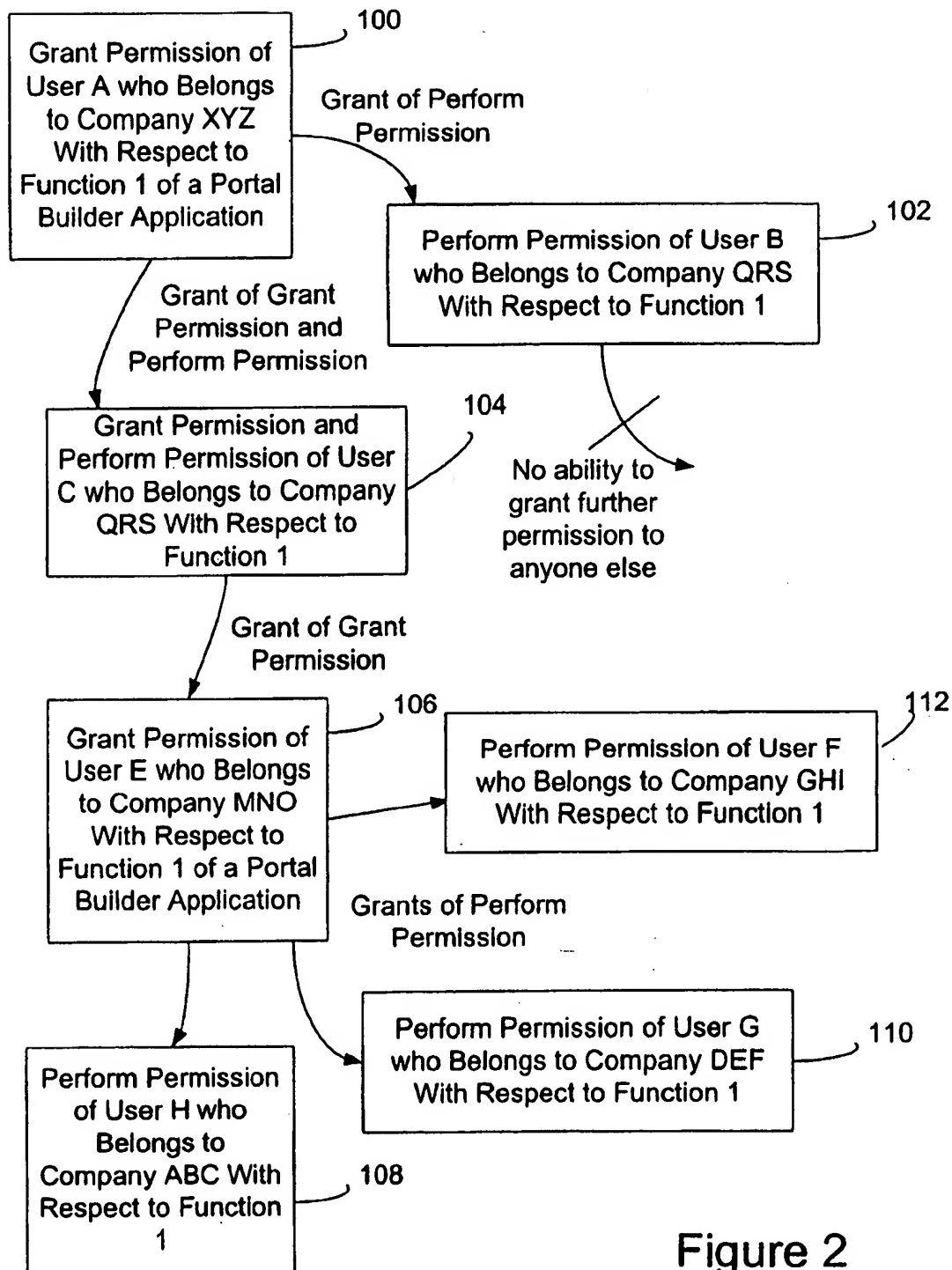


Figure 2

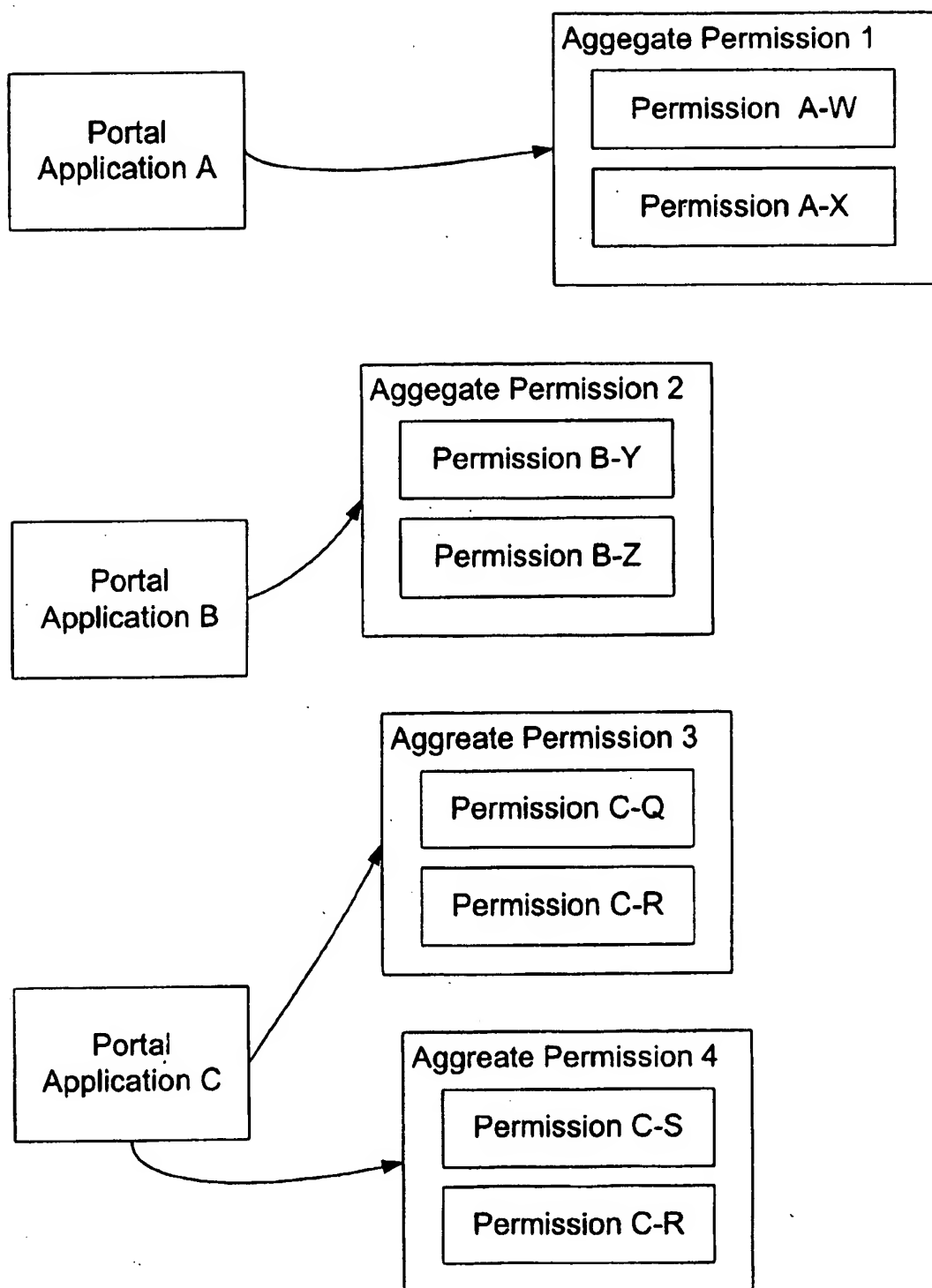


Figure 3

MANAGING RELATIONSHIPS OF PARTIES INTERACTING ON A NETWORK

BACKGROUND

This invention relates to managing relationships of parties interacting on a network.

A party that has information to be made available to other parties with which it has or wishes to have relationships can disseminate the information to the other parties on a web site using a web server that is accessible by web browsers.

In a commercial context, for example, a manufacturer may try to increase sales of its products or services by giving better information and support to resellers or other companies that are in the chain of distribution between the manufacturer and end customers. On the other hand, the manufacturer may want to screen a particular type of customer from having access to information that is not targeted to that type of customer.

Access to the information by different parties can be regulated by firewalls, by password techniques, and in other ways.

SUMMARY

In general, in one aspect of the invention, at least one digital facility is made available via an electronic communication medium. Information is stored that reflects the existence of relationships between identified parties with respect to use of the digital facility. A predetermined type of interaction of the parties is permitted via the electronic communication medium with respect to the digital facility if the stored information reflects the existence of relationships between them.

Implementations of the invention may include one or more of the following features. The stored information identifies the nature of interaction permitted or precluded between identified parties with respect to use of the digital facility. The nature of interaction includes one of the parties being exposed to the existence of another of the parties in connection with using the digital facility. The existence of the other of the parties is made apparent by inclusion of the other of the parties in a displayed list of parties with whom interaction is permitted, the displayed list being determined by the stored relationship information. The interaction of parties via the electronic communication medium is governed by permissions defined with respect to the relationships. The permissions include a permission to be aware of the existence of specified other parties. The parties include individuals, groups of individuals, and commercial enterprises. The permitted interaction includes working together on a task, delivery of content, or one party accessing specified data that is associated with another one of the parties. Individuals may have a relationship only if respective institutions with which they are affiliated have a relationship. The stored information is created and controlled by parties only in accordance with permissions belonging to those parties. The predetermined types of interactions that are permitted and precluded are defined in stored permissions.

Among the advantages of the invention are one or more of the following. The system enables the portal-providing company and other companies interacting through the portal to make better use of their marketing and sales people and resources. The portal-providing company and other companies interacting through the portal can use the system to

motivate their selling partners to achieve higher sales. The impact and reach of the portal-providing company's direct sales people and the front-line teams of other companies interacting through the portal can be improved. Selling partners and customers develop a greater awareness of the portal-providing company and other companies interacting through the portal. Selling partners are given enhanced marketing capabilities. The portal-providing company and other companies interacting through the portal can present a single coordinated image to selling partners and customers while providing custom experiences for individual users. The system also enables improvement of the performance of many other business processes that occur between and within companies. These improvements may take the form of reduction in the time required to execute. The system can improve the performance of interactions with a company's suppliers and vendors. It can also improve the performance of processes that occur after a business transaction, including the execution of implementation projects, service, and support. And it can improve the performance of a wide range of internal company processes.

Other advantages and features will become apparent from the following description and from the claims.

DESCRIPTION

FIG. 1 is a block diagram of a relationship management system.

FIG. 2 is a schematic diagram of examples of the granting of perform and grant permissions.

FIG. 3 is an illustration of aggregate permissions.

OVERVIEW

As shown in FIG. 1, a portal-providing company 20 can use relationship portal software 21 running on its web server 22 to leverage the value of information stored in its business database 24. Software 21 can be used to regulate relationships that the portal-providing company has with other parties 26, including the portal-providing company's direct customers 28, other companies 30 along a chain of distribution 32, end customers 34, and marketing partners 36. The software can also be used to regulate relationships that parties in the chain other than the portal-providing company have with each other (e.g., a relationship between a direct customer and an end customer), at least to the extent that the relationship involves the portal-providing company's information.

The business database 24 can be an existing database or a newly created database. Typically the business database would include information about products or services, prices, terms, availability, marketing plans, historical and projected sales, and any other information that would be useful in facilitating the relationships among the companies and employees who are authorized to use the system.

The regulation of relationships by software 21 can include control of which information of the various companies that participate in the portal environment is made available to which other companies and to which employees of those companies. The regulation can also include control of how the information is presented to the employees of the companies to enable the delivery of different "experiences" to different companies and their employees. In that way, the software 21 and the web server 22 can provide a personal relationship portal 27 that can have custom appearances and behaviors for each of the employees that communicates with the web server through a web browser. Each employee of

3

any of the companies can get access to the web server from anywhere in the world through a network 40 that could be, for example, the Internet, an Intranet, a local area or wide area network, a dial up connection, or any other arrangement that allows communication between the employee's web browser and the network.

Software 21 provides two main functions: configuration of the software to serve a portal-providing company and its relationship partners, and run-time operation of the software to provide the personal relationship portals to employees of the companies.

The configuration function includes identifying and creating profiles for companies and employees of companies who will have access to the system, and defining the relationships among the companies and employees, the rights that each company and employee will have to use the system, and the preferences of each employee that together establish the nature and content of each employee's personal relationship portal. The configuration information is kept in a portal management database 23 associated with software 21.

Run-time operation includes conveying each of the personal relationship portals to web browsers in accordance with the configuration and information contained in the portal management database 23 and in the business database 24. In addition to providing information, each personal relationship portal can provide functions and services to the employee using the browser.

A wide variety of hardware and software can be used to implement the web server 22. The relationship portal software 21 and the portal management database 23 may be based on an appropriate set of platforms that can include Microsoft Transaction Server (MTS) and Microsoft Internet Information Server (IIS). The relationship portal software may take advantage of typical web software capabilities including Active Server Pages (ASP), Component Object Model (COM) objects, Java, Extensible Markup Language (XML), and ActiveX Database Objects (ADO). In one implementation, the relationship management software comprises a Java class library that implements a suite of portal applications 29. The portal applications provide both the configuration capabilities and run-time features of the system. Other platforms may be used in implementations of the invention, including BEA Systems WebLogic Server, IBM WebSphere Application Server, Java Server Pages (JSP), Enterprise Java Beans (EJB) and Java Database Connection (JDBC).

The relationship portal software may include a generic schema for the portal management database 23. The database initially includes seeded descriptive data in the form of query scripts. The Java class library can "talk to" the database, i.e., gather data from the database, manipulate the data, and store data in the database. The relationship portal software includes HTML web pages and/or can include ASPs that can communicate with the Java class libraries using COM communications. The server distributes the resulting web pages in the form of personal relationship portals to the user browsers.

The relationship portal software handles two types of parties: users and companies. A user is a person who is employed by or is otherwise associated with ("belongs to") a company. Information about users, groups of users, and companies is generated as part of the configuration process. The information is stored and maintained as profiles in tables in the portal management database 23.

The profiles include two important classes of information, called permissions and preferences. The permissions define

4

the rights of users and companies with respect to information and functions that are made available by the portal applications. Permissions are granted by users of the system in ways (described below) that enable the development of a rich, distributed fabric of personal relationship portals for a range of users and companies who are in the constellation of interest of the portal-providing company. All of the personal relationship portals are implemented from the portal-providing company's server, but the particular permissions and preferences of any one of the personal relationship portals are not necessarily fully controlled by the portal-providing company. Rather the ability to control them is distributed so that other companies within the constellation and other users in those companies may be able to influence the configurations and permissions associated with some of the portals. How this is done will be explained more fully below.

Preferences define how the user wishes to receive the information and services made available by the portal applications. Permissions and preferences are similar in that they are unique named objects defined by a portal application. Actual permission values for a user represent specific access rights to application functionality, whereas specific preference values influence application behavior.

USER PROFILES AND COMPANY PROFILES

It may be required that a profile for a company to which a user belongs must exist before a profile for the user can be created.

A company profile can include demographic data; lists of other companies with which the company has a relationship; in some implementations a list of the maximum permissions that users at the company can be given in creating or managing the profiles of other users and companies; and a list of other users who are allowed to manage the profile of the company. A user profile may include demographic data; permissions given to the user to create and manage the profiles of other users and companies; a list of permissions given to the user with respect to other objects in the system; a list of preferences of the user for his personal relationship portal; and a list of other users who are allowed to manage the profile of the user.

The user profile contains a listing of permissions to which the user has some level of perform permission (defined below). In one implementation, the maximum permissions available to a user are limited by the permissions assigned to the user's company. Once permissions are assigned to a user, the company permissions are not consulted in order to determine what facilities are available to the user. However, when company permissions are removed, they are also removed from all of its employees.

Once a user has been created, the "belongs to" relationship of that user to his company cannot be changed except to the extent that the user wanting to make the change has permission to do so.

A company profile can be created and exist independently of any other company or user. This conveniently enables company profiles to be created and to exist without users until a later time.

For users who are at the end of a relationship chain (e.g., who don't work for a company), such as shareholders, an artificial company profile in the name of, e.g., "Shareholders", could be created. The Shareholders company would be simply a grouping mechanism for those users.

Each user profile or company profile has a "profile manager". The profile manager is the user who has the

5

permission to change settings and possibly delete the profile. These abilities are expressed in the "modify user profile" and "remove user profile" permissions. One or more aggregate permissions (discussed below) may be configured to set the manager permissions.

A person who has permission to, and who does initially create a user is automatically the profile manager for the new user. The profile manager will have both perform and grant permissions of the aggregate permissions. The relationship between the permission to create an object and the permission to manage an object is configurable. The concept of managing an object derives from aggregate permissions, which are discussed below. Aggregate permissions often include fundamental permissions to view, modify and delete an object, which are aggregated into what is effectively a right to manage that object.

User profiles are created by other users. In another implementation, users are given permission to manage all users at particular companies and there are not user management permissions. A company profile is created by a user. The user who sets up a company profile automatically becomes a manager of that company's profile.

Some users may be allowed to create companies of certain types. Company types are themselves configurable and may differ from one portal installation to another. The users automatically become managers of the companies they create. Other users may be given permission to manage certain company profiles but not to create any. These company managers can change demographic data freely, but they can only change permissions to which they themselves have the grantright.

GROUPS

Permissions may also be assigned to users through the mechanism of groups. A group is a collection of users. A user with a grant right to a permission may give that permission to a group just as he or she would give it to an individual user. Once the permission has been given to a group it is automatically conferred on all members of that group. Users may become members of groups in two ways. First, a user who has the permission to manage a group can designate members individually. Second, a user who has the permission to manage a group can specify that users with a certain demographic profile will automatically be members of that group. Any user who becomes a member of a group immediately acquires all the permissions that have been given to the group. The use of groups simplifies administration of permissions by enabling a large number of permissions to easily be assigned by putting users in just a few groups.

PORTAL BUILDER APPLICATION

The configuration functions are provided by one of the portal applications, called "Portal Builder". Portal Builder enables users to view, modify, remove, and create user profiles and company profiles and in that manner to set the stage for personal relationship portals that are made available at run-time to users of the system. Each user has access to Portal Builder to view and modify portions of his own user profile. This capability is limited by the user's permissions with respect to him.

Specific functions of the Portal Builder, such as creating, modifying and removing user profiles, are available to a user only if the user has permission to access those functions. Typically those functions are used by individuals identified as relationship managers to configure personal relationship portals for customers of their companies.

6

Portal Builder displays web pages to present and collect the demographic data about users and the companies they work for and to establish user permissions and preferences defined by portal applications. As a portal application, Portal Builder defines its own set of permissions that control how it can be used and that it also manages when configuring a personal relationship portal.

A portal application uses the Portal Builder to define permissions that must exist for a user to make use of the application. The Portal Builder assumes the responsibility for gathering that information on behalf of the other portal, applications.

The table set forth in appendix A lists the fundamental components of the profile service of the Portal Builder in one implementation. A user interacts with the Portal Builder through a series of web pages that walk the user through a process. The screens associated with the web pages for one implementation are listed in the table of Appendix B.

PERMISSIONS

The ability of a user to access the functions and information available through the relationship portal software 21 is governed by a system of permissions. The experience of a user on his personal relationship portal is determined in large part by the permissions that he has. The permissioning architecture is highly adaptable and highly granular. The identity and scope of the permissions to which a user is entitled are determined by other users and are restricted by a set of rules that limit the ability of a user to grant permissions to other users. If carefully constructed, the fabric enables a portal-providing company, and other companies who participate in the system, to control the dissemination of information and the access to services in a way that enhances and facilitates their relationships with parties who can advance their interests. Conversely, dissemination of information and the access to services can be restricted for parties who can injure the interests of the portal-providing company.

A permission specifies the right that a user may be given to access the functions of a portal application. Roughly speaking, a permission is a capability to do "something" (through the medium of the portal application), perhaps with respect to "something else". For example, the something which a permission enables may be to change a profile of a company, or to modify a user profile, or to create a user, or create a company of a particular type. A permission has an argument, which refers to the "something else" to which the capability pertains. The argument can refer to any object in the system such as a company or a piece of data.

Each portal application 29 defines permissions that a user must have in order to access the application or specific functions of the application. The application defines the arguments of any permissions that it defines. A user may not access features and functions of portal applications unless the user has a "perform" permission (defined below) with respect to the feature or function. At run-time, the portal application uses permission information maintained by the Portal Builder to determine whether and how to respond to a user request.

The definitions of possible permissions are stored in a permission definition table in the portal management database. The actual specific permissions associated with a given user or company are stored as values in a party permission table of the portal database. Portal applications that are invoked by a user consult the party permission table to determine if a user has a needed permission.

7

A permission may (but need not) have an argument, a value specifying an entity to which the permission applies. For example, a permission to create a company carries an argument that identifies the type of company.

In one implementation, when the database table of permissions is initially created, there is one set of permissions granted to one person called the system administrator. These permissions give the system administrator the right to create and modify companies of all types. In this implementation, the system administrator company (a fake company whose sole employee is the system administrator) is also given this permission. The system administrator and company are also given other global permissions with respect to other applications. In another implementation, the system administrator is given a "wildcard" argument value to all permissions, which translates to the complete list of possible objects for each permission.

COMPANY PERMISSIONS

In some implementations, permissions apply both to companies to which a user belongs and to the user himself. Although only individuals are direct users of relationship portals, permissions may be recorded for companies to which the people belong to enable further regulation of the permission fabric. The permissions that can be given to a user are constrained by both the permissions of the company to which he belongs and by the permissions of the user who is granting the permissions to the recipient. The permissions of a company (also called guard settings) define the maximum amount of permissions that a user who belongs to the company can be given. A user, creating a new company asks, "What are the maximum amount of permissions that any user at this company may need?" and then gives the company those permissions. If a user who belongs to that company later needs more permissions, new company permissions must be added before the user can be given the permissions.

PERFORM PERMISSIONS; GRANT PERMISSIONS; CASCADING PERMISSIONS

A permission may specify either or both of two aspects of permission: a perform right and a grant right. The perform right of a permission (the "perform permission") gives a user the right to perform a specific function offered by a portal application. As suggested in the example shown in FIG. 2, the grant right of a permission (the "grant permission") gives a user the right to grant to other users the perform permission and the grant permission with respect to the application functionality associated with the permission. The grant and perform permissions are represented by flags in the profiles stored in the portal management database. As shown in FIG. 2, user A's grant permission 100 enables user A to grant perform permission 102 to user B and grant permission and perform permission 104 to user A user (such as user B) who does not have the grant permission cannot modify the perform permission for any other user, either to add it or to remove it. A user who has the grant permission may give the perform permission or the grant permission or both to another user. In this way, the "cascading" of permissions can continue without limit as long as each user gives grant permissions to next users in the cascade. In FIG. 2, user C can give grant permission 106 to user E who works in another company and user E can then give perform permissions 108, 110, 112 to three other users F, G, and H in three other companies.

The result can be cascading tiers of personal relationship portals. For example, a selling partner or customer of the

8

portal-providing company can create personalized branded web sites for its customers or downstream partners and can reuse information and services available on the portal-providing company's portal.

A user (such as user A) can have the grant permission without having the perform permission. This is natural, because a user may be responsible for managing the permissions of users who have responsibilities that the manager himself does not have. For example, a sales person may be responsible for managing the permissions of all users at an account. One of those users may be a lawyer. The sales person must be able to set up the lawyer's permissions but should not have access to the legal content. Although it would be possible for a malicious user to create a fake user and log in as the fake user to get access to functions not meant for him, that act would create an easily traceable trail.

CASCADING PERMISSIONS AND USER/COMPANY RELATIONSHIPS

The link between cascading permissions and the relationships of users to companies is suggested by what happens to user permissions when a change occurs for the company to which the user belongs.

For example, suppose that a company is initially defined in the portal management database as a "Tier I" company (e.g., a large or highly profitable company). The portal-providing company wishes to permit Tier I companies to have only the ability to submit questions to an Ask the Expert application, and companies of all other Tiers to have only the ability to read the questions and answers.

If the Tier I company becomes less profitable and falls beneath the Tier I threshold, the company and its users will need to be switched into the Tier II category. The effect of the switch should be to remove from users who belong to the company the ability to submit questions to Ask the Expert and leave them only with the ability to read.

Conversely, suppose that a company is initially recorded as a "Tier II" company. Because the portal-providing company only wants Tier I companies to have the ability to submit questions to the Ask the Expert portal application while giving all other Tiers only the ability to read, this company may only read postings to Ask the Expert.

If the company becomes more profitable, the company can be switched to Tier I in the database to give its users the ability to submit questions to Ask the Expert.

When additional permissions are added to a company, the user who is making the change to the company record can pick and choose manually which users of the company should have their permissions changed. When permissions are removed from a company, all users belonging to that company should lose the removed permission.

FUNDAMENTAL PERMISSIONS

Examples of basic permissions (called "fundamental permissions") that a portal application might define and invoke are shown in the table contained in Appendix C.

Detailed explanations of four possible permissions follow.

Create Company—This permission gives the user the ability to create new company profiles in the portal management database. A user who has this permission is automatically given the remaining permissions, described below, for a company that he creates.

Modify Company—This permission gives the user the ability to modify the permissions of existing companies. The permission includes the permission to modify the permis-

sions of all users belonging to that company and to create new user records belonging to that company. The argument of this permission is the list of companies that the user may be allowed to modify. The grant permission of this permission specifies which companies the user can give other users the ability to modify.

Create Company User—This permission gives a user the ability to create other users. The argument of this permission is the list of companies at which this user may create users. The grant component of the permission specifies the companies at which the current user may enable other users to create users.

Modify Company User—This permission enables a user to modify the permissions of specific users. The argument of this permission is the list of companies at which this user may modify users. The grant component of the permission specifies the companies at which the current user may enable other users to modify users.

AGGREGATE PERMISSIONS

Because the fundamental permissions are so granular, it is also convenient to provide an "aggregate" permission capability. As shown in FIG. 3, an aggregate permission 1, 2, 3, or 4 is defined by the portal application A, B, or C that may invoke the permission at run-time. An aggregate permission is not a permission by itself but only a defined aggregation of permissions. The fundamental permission definitions are stored in one table in the portal management database. The aggregate permission definitions are stored in another table, and the mappings between the aggregate permissions and the fundamental permissions are stored in a third table. The permission values, the actual settings for a person or company, are stored in yet another table. This table includes the fundamental permission value and the aggregate it is associated with. The reason for this is that in the case of overlapping aggregates (see below), the same fundamental permission may have different values. The rule is that the value with the greatest rights wins. When one of the overlapping aggregate permission values is removed, the fundamental permission values associated with the other overlapping aggregate permission is retained.

Every aggregate permission is associated with one portal application. The aggregate permission 1 incorporates permissions A-W, A-X, and so on, in each case only for the portal application with which it is associated. A portal application C, for example, may be associated with more than one defined aggregate permission 3 and 4. Aggregate permissions may overlap, meaning that they map to the same underlying, fundamental, permissions.

Users of the Portal Builder who are creating or modifying user permissions may work only at the aggregate permission level, not at the individual application permission level. A user profile is defined by one or more selected aggregate permissions, and the users only need to know how to select aggregate permissions for users. A user who has a relationship manager responsibility will dispense that responsibility at the aggregate permission level.

Each aggregate permission is given a name that describes, for example, a form of business function, usually in the context of the portal application that defines that permission. Portal application permissions that need to be accessible to someone whose role is to fulfill the business function are enabled with the perform aspect of the aggregate permission.

Certain portal applications that may be considered critical services are always enabled for every portal user. This can be done by providing a mechanism to enable permissions of

those applications without use of the aggregate permissions model and without requiring user action.

Setting the perform or grant rights of an aggregate permission is equivalent to setting those rights for all of the constituent fundamental permissions.

Aggregate permission definitions are stored in an aggregate permission database table. The table is initialized when a portal-providing company's portal is installed, but unlike fundamental permissions, aggregate permission definitions can be modified without any corresponding change in applications that rely on the permissions that are part of the aggregate permission. However, the definitions are not modified after the portal is installed. Aggregate permissions make it easier to modify the configuration portion of an application without modifying the underlying application functionality. For example, new aggregate mappings can be installed in the database, and new screens can be created to set these aggregates, all without changing the application code that consults fundamental permissions to see what actions are allowable. Nevertheless, all of these definitions are put in place before a portal is installed.

EXAMPLE OF AGGREGATE PERMISSIONS

Assume that the fundamental permissions include:

Create Company, argument is company type. User can create companies of the specified types.

Modify Company, argument is company. User can modify the profiles of the specified companies.

Delete Company, argument is company. User can delete the profiles of the specified companies.

Create Company Users, argument is company. User can create users at the specified companies.

Modify Company Users, argument is company. User can modify the profiles of users at the specified companies.

Delete Company Users, argument is company. User can delete the profiles of users at the specified companies.

Assume that the aggregate permissions include:

Company Creator, argument is company type. Maps to Create Company and functions the same way.

Company Manager, argument is company. Maps to Modify Company, Delete Company, Create Company Users, Modify Company Users, and Delete Company Users. User can modify and delete company profiles, and create, modify and delete users profiles at the specified companies.

Company User Manager, argument is company. Maps to Create Company Users, Modify Company Users, and Delete Company Users. User can create, modify and delete users at specified companies.

Assume that the system administrator and company have perform and grant permission for all values of Company Creator aggregate.

When the system administrator creates company A, he automatically becomes the Company Manager and Company User Manager of company A. He gives company A perform and grant rights to the Company Manager and Company User Manager aggregate permissions for company A. All users at company A now have the potential to manage the company profile and the profile of all users at the company.

If system administrator creates user B at company A and gives him the full set of permissions allowed to company A, user B can manage the company profile, and create and manage all user profiles at the company, but he cannot create any new companies.

11

If user B creates user C at company A and gives her perform rights to the Company User Manager aggregate permission for company A, then user C can create and manage the profiles of all users at company A, but cannot manage the company profile itself, and cannot give anyone else the right to manage user profiles at the company.

USER PREFERENCES

A user's profile includes preferences, which capture how the user wants to set up his personal relationship portal and to receive information. The preferences that are available to a user are limited by the user's permissions.

A portal application may specify preferences that generally control how a feature will operate within the portal application, provided the user has the permission to use that feature. An example of a preference is a selection of seven of ten available news channels to be displayed in a news application. The permission is the ability of the user to access the news application; the preference is what news sources will be displayed.

Preferences are only associated with users, not with companies, because preferences are more closely related to how a user makes use of a user portal.

Preferences are specific to one user only. Preferences cannot be cascaded. When a user is granted permission to access a portal application, the application automatically gives the user access to all relevant preferences.

For example, suppose that preferences "sort order" and "filter by" pertain to view and modify permissions of a particular portal application and that the preference "confirm before delete" pertains to the "remove" permission for that portal application. If a user does not have access to the "remove" permission, when the preferences page is generated for a user, the portal application is coded not to display the "confirm before delete" preference.

When a user is initially created, his preferences will default to a standard setting, regardless of the user type or company. Unless the user performing the user creation or modification purposely chooses to modify the user's preferences, the preferences remain in the default state.

Preference templates simplify the process of configuring a user's preferences. A preference template specifies the settings for a number of individual preferences. When a preference template is applied to a user, all of the user preferences are set as specified in the template. Preference templates can be applied by users to their own profiles, or they can be applied by a user's profile manager to that user's profile on his or her behalf.

Preference definitions are recorded in a table in the portal management database that includes, in one implementation, the name of the preference, a label for the preference, a flag indicating whether it accepts single or multiple values, a type code indicating whether the value is accepted as an input field, list, or set of choices, one or more default values for the preference, a set of values for the list elements or choices, an optional set of labels for the list elements or choices, and an optional set of default values for the list or choices (one value for an input field).

Actual preference values are stored in another database table in the portal management database.

OTHER IMPLEMENTATIONS OF RELATIONSHIP MANAGEMENT

In another implementation approach, companies may be defined as belonging only to specified types that correspond,

12

for example, to their roles in a commercial supply chain, and each user can be defined as having only a specified role within the company to which he belongs. For example, the company types could include sales representatives ("reps") and suppliers who are represented by the manufacturers ("principals"). Within a rep company, a user could have a role as, for example, an outside sales rep, an inside sales rep, or a rep manager. Within a principal company, a user could have a role as a product manager or national sales manager, for example.

Relationships may be defined between parties of specified types and between individuals having specified roles in respective companies. For example, a product manager of a supplier may have a relationship with a sales representative of a vendor but may be denied a relationship with a product manager of the vendor. And individuals who belong to a supplier may not be permitted to have relationships of any kind with any individuals who belong to another supplier. In addition to companies and individuals, the system can control relationships between groups of individuals, for example, the client team for client X in company Y and a quality review team for vendor Y in client X.

The relationships can be used to govern the ability of the individuals, groups, and companies to access and use content and features maintained on a relationship server.

Among the content and functions that could be provided by the server in a given implementation are a new business application used to manage and track new business opportunities, requests for samples, action item tracking, electronic discussions, registration of companies and users by the setting up of profiles, an address book, and administrator functions that enable a designated administrator for a given company to control the creation and maintenance of user profiles and relationships and a designated system administrator for the portal-providing company of the server to control the creation and maintenance of company profiles.

The user interface that is presented through a browser to a given user depends on the user's role, on user and company profiles, and on definitions of relationships between individuals, groups, and companies. Among the facilities provided on various user interfaces are the ability to invite participation by users at other companies with which the company to which a user belongs has a relationship, to create a document, to manage (create, view, modify, and delete) commercial opportunities, to manage (create, view, modify, and delete) requests, to manage (create, view, modify, and delete) action items, to manage (create, view, modify, and delete) address books, and to control user preferences.

A company administrator also has user interface access to facilities for approving new users, approving new relationships, managing relationships, requesting relationships, altering a company profile, and viewing users. The overall system administrator is provided with user interface access to the ability to view companies and users.

For each function, each role of a user, and each company type to which the user belongs, the system defines permissions that determine whether a user having that role can access that function. The user interface is constructed to enforce the defined permissions.

The relationships between companies and between users are enforced by the implementation of the user interface based on the stored relationship information. For example, if a user does not have permission to view a specific company or a specific user, then the various lists and other interface elements that display to the first user companies and users

13

with which he can interact will not include the non-permitted parties. In effect, the first user will have no way to know that the non-permitted parties even exist let alone any way to interact with them through the system. Thus, the user interface regulates and implements the permitted relationships for each user and company automatically based on the stored permissions.

Relationships are established as follows.

A new company may initiate a registration by providing appropriate fields of information for a profile through the user interface presented by the browser. The system administrator will review the information and either approve or disapprove the registration. The user who initiated the registration process for the company becomes the company administrator.

A user profile for a registered company is also created through the user interface presented by the browser. After the new user initiates the registration process by entering information needed for his profile, the record is presented to the company administrator who can approve or disapprove it.

Because the company profile and the user profile define the company type and the role of the user, the permissions applicable to such companies and users can be used to regulate the relationships. For example, the permissions may provide that a national sales representative of a rep firm can have a relationship with a product manager of a principal company with which his company has a relationship. Then, when a new product manager is added to a principal company, the new product manager automatically acquires the relationship permissions associated with new product managers of principal companies. The role that a user has within a company is controlled by the company administrator.

Setting up a relationship between one company and another (or between other parties such as individual users or groups) is initiated by the one company or party requesting the relationship with the other. The company administrator of the other company is presented with the request and decides whether to approve or reject the request. If approved the relationship is recorded on the system. As part of the approval process, the administrator is able to view a list of users who have certain roles for the company seeking the relationship. In a variation on this implementation, each party is assigned a "key" (a code sequence known only to authorized users). Any user who knows the key can use it to establish a relationship with the party to which the key applies. Transmittal of the key occurs through means outside of the portal. In this way, users that have an existing (or desired) business relationship can establish a portal relationship before they are able to use the portal as a communication mechanism between them.

In another implementation, the permission to establish a relationship between companies (or other parties such as individual users or groups) belongs to any user who is a manager of both the parties that will be involved in the relationship. In this case, parties cannot initiate the relationship building process themselves; it requires a third party.

The effects of the relationships between users and companies are determined by the types of companies and the roles that the individuals play in those companies. Permissions are applied automatically through relationship definitions. For example, a chain of people may have a relationship with respect to assembling and delivering content, for example, a product brochure. The same people may have no other relationship to one another with respect to a different set of activities.

14

Permissions can be driven based on attributes of a relationship. If each user's profile is defined to include demographic attributes such as geographical location, the right to see highly confidential information, and existing relationships with specified other parties, then another party can specify, for example, that he wishes to permit users who have certain demographic characteristics to receive a certain piece of marketing literature. In such a system, the relationships need not be governed by predefined roles, or by companies or groups to whom the users belong, or by pre-specified permissible relationships.

Other embodiments are also within the scope of the following claims. For example, the mechanism by which the user interacts with the system need not be a web browser. The user interaction can be governed by devices other than a web server. The parties who make use of the system need not be companies and their employees, but could be any entities and individuals who "belong" to them. The context in which the invention is implemented need not be based on business relationships, but could be based on any relationships that entities may wish to regulate.

Appendix A

Component	Type	Description
AggregateElement	Database table	Contains mappings between aggregate and fundamental permissions
AggregateElement	Java class	Manages the fundamental permissions associated with an aggregate permission
AggregatePermission	Database table	Contains aggregate permission definitions
AggregatePermission	Java class	Manages data associated with aggregate permission definitions
AggregateValue	COM object	COM wrapper for business logic
AggregateValue	Java class	Manages setting aggregate and fundamental permission values
PartyPermission	Database table	Relates Party objects to a set of permission values
PartyPreference	Database table	Relates Party objects to a set of preference values
PartyPreference	Java class	Manages data associated with preference values
PermissionValue	Java class	Retrieves individual permission values
PortalPermission	Database table	Contains application permission definitions
PortalPermission	Java class	Manages data associated with permission definitions
PortalPreference	Database table	Contains application preference definitions
PortalPreference	Java class	Manages data associated with preference definitions
PreferenceChoice	Java class	Manages data associated with preference choices
PreferenceDisplay	Java class	Formats permissions into XML for display
PreferenceManager	Java class	Manages the interaction between preference defaults and values

Appendix B

Screen Name	Screen Description
Start Menu: "My Preferences"	This is actually not a screen but a component of the Start Menu. This is the component that an end-user uses to access personalization features such as

-continued

Screen Name	Screen Description
Start Menu: "Portal Builder"	preference settings and access to demographic information contained in his/her profile. This is actually not a screen but a component of the Start Menu. This is the component that contains administrative features such as access to company and user profiles, sending password reminders, and managing templates.
Select User	This is an implementation of a standard selection screen. It allows a user to examine the full list of users s/he may access and select one for modification, removal, or choose to create a new user.
Common User Demographics	A data collection screen that every portal user is required to fill out. This is also the screen that allows the portal password to be set and changed.
Job Function Additional Data Collect	This screen contains a dynamic selection of data elements depending on the Job Function(s) selected for a user. The purpose of the screen is to capture data elements unique to the Function the user plays. This screen displays all aggregate permissions the user being modified may be given access to.
Select Aggregate Permissions (user profile) Application & Permission On-line Report Configure Application Preferences	This screen displays to the user what applications and permissions are available to the user as a result of the selected aggregate permissions. Allows for the configuration of application preferences. Only those applications to which the user has permission to use will be available for preference setting.
User Managers	This screen appears as part of user creation/modification and is used to identify what other portal users may update this user's profile.
Company Select	This is an implementation of a standard selection screen. It allows a user to examine the full list of companies s/he may access and select one for modification, removal, or choose to create a new company.
Common Company Demographics Company Type Additional Data Collect	A data collection screen that every portal company will be required to fill out. If additional data is necessary based on the Company Type selected, this screen is generated to collect those elements.
Select Aggregate Permissions (company profile) Company Organization	This screen allows for the selection of aggregate permissions. These permissions define what this user may do in the portal environment. This screen allows a company to define its internal organization (e.g. regions, branches, offices, etc.) This information is used as an attribute of each user and might be leveraged by portal applications.
Company Managers	This screen appears as part of company creation/modification and is used to identify what other portal users may update this company's profile.
Company Profile Completed	This screen marks the end of a company profile creation or modification. Options exist here to determine how users at that company should be affected.
Password Reminder	Using the standard user selection screen, this form allows users to be selected and an e-mail sent to their e-mail address on file with a password reminder.

Appendix C

Permission	Argument	Description
Create Company Type	Company Type	Create companies of the specified type
View Company Type	Company Type	View companies of the specified type
Modify Company Type	Company Type	Modify companies of the specified type

-continued

Permission	Argument	Description
5 Delete Company Type	Company Type	Delete companies of the specified type
View Company	Company	View specified company
Modify Company	Company	Modify specified company
Delete Company	Company	Delete specified company
Create Company	Company	Create users at specified company
10 User		
View Company User	Company	View users at specified company
Modify Company User	Company	Modify users at specified company
Delete Company User	Company	Delete users at specified company
15 View User	User	View specified user
Modify User	User	Modify specified user
Delete User	User	Delete specified user

What is claimed is:

1. A method comprising making at least one digital facility available via an electronic communication medium, storing information that reflects the existence of relationships between identified parties with respect to use of the digital facility,
- 25 storing information that reflects preferences of the identified parties, and permitting a predetermined type of interaction of the parties via the electronic communication medium with respect to the digital facility based on the stored information that reflects the preferences of the identified parties and the stored information that reflects the existence of the relationships between the identified parties.
- 30 2. The method of claim 1 in which the stored information identifies the nature of interaction permitted or precluded between identified parties with respect to use of the digital facility.
3. The method of claim 2 in which the nature of interaction includes one of the parties being exposed to the existence of another of the parties in connection with using the digital facility.
4. The method of claim 3 in which the existence of the other of the parties is made apparent by inclusion of the other of the parties in a displayed list of parties with whom interaction is permitted, the displayed list being determined by the stored relationship information.
- 45 5. The method of claim 1 in which permitting the interaction of parties via the electronic communication medium is governed by permissions defined with respect to the relationships.
6. The method of claim 5 in which the permissions include a permission to be aware of the existence of specified other parties.
- 50 7. The method of claim 1 in which the parties comprise individuals.
8. The method of claim 1 in which the parties comprise groups of individuals.
9. The method of claim 1 in which the parties comprise commercial enterprises.
- 60 10. The method of claim 1 in which the permitted interaction includes working together on a task.
11. The method of claim 1 in which the permitted interaction includes delivery of content.
- 65 12. The method of claim 1 in which the permitted interaction includes one party accessing specified data that is associated with another one of the parties.

17

13. The method of claim 1 in which individuals may have a relationship only if respective institutions with which they are affiliated have a relationship.

14. The method of claim 1 in which the parties comprise commercial entities belonging to different levels of a supply chain.

15. The method of claim 1 in which the parties comprise groups of individuals and the relationships include relationships between two of the groups or between a group and the company.

16. The method of claim 1 in which the stored information is created and controlled by parties only in accordance with permissions belonging to those parties.

17. The method of claim 1 in which the predetermined types of interactions that are permitted and precluded are defined in stored permissions.

18. An apparatus comprising:

a plurality of applications;

a portal management database comprising

a plurality of fundamental permissions,

a plurality of aggregate permissions, each aggregate permission including one or more fundamental permissions, each aggregate permission associated with one of the applications,

a first group profile for a first group, the first group profile including a plurality of first group permissions based on a mapping of the aggregate permissions with the first group,

a plurality of first group member profiles for a plurality of first group members, each first group member profile including a plurality of first group member permissions based on a mapping of the plurality of first group permissions with the first group member.

19. The apparatus of claim 18, further comprising:

a plurality of personal relationship portals, each personal relationship portal to provide one of the first group members with access to at least a portion of a given application based on the first group member permis-

18

sions and the aggregate permissions associated with the given application.

20. The apparatus of claim 18, wherein the plurality of fundamental permissions further comprises a perform permission to permit a member to perform at least a portion of one of the applications.

21. The apparatus of claim 20, wherein the plurality of aggregate permissions further comprises a grant permission to permit a member to grant a permission to one or more first members.

22. The apparatus of claim 20, wherein the plurality of aggregate permissions further comprises a perform permission to permit a member to grant a perform permission to a portion of the first group members excluding the granting member.

23. The apparatus of claim 18, wherein the plurality of fundamental permissions further comprises permissions to permit a member to create or modify the first group profile or to create or modify another first member profile.

24. The apparatus of claim 18, each first group member profile to include a list of preferences related to the first group member.

25. The apparatus of claim 18, the portal management database further comprising a second group profile for a second group.

26. The apparatus of claim 25, the portal management database further comprising a plurality of second group member profiles for a plurality of second group members, the second group profile to include a plurality of second group permissions and a plurality of second group member relationships, each second group member relationship to relate one of the second group permissions to one or more second group members.

27. The apparatus of claim 25, the first group profile to further include a first group relationship to relate one of the first group permissions to the second group.

* * * * *



US006516315B1

(12) **United States Patent**
Gupta

(10) Patent No.: **US 6,516,315 B1**
(45) Date of Patent: **Feb. 4, 2003**

(54) **METHOD FOR CONTROLLING ACCESS TO INFORMATION**

(75) Inventor: **Arun Kumar Gupta**, Easton, CT (US)

(73) Assignee: **NeuVis, Inc.**, Shelton, CT (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/434,895**

(22) Filed: **Nov. 5, 1999**

Related U.S. Application Data

(60) Provisional application No. 60/107,126, filed on Nov. 5, 1998.

(51) Int. Cl.⁷ **G06F 17/30**

(52) U.S. Cl. **707/9**

(58) Field of Search **707/9**

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,621,321 A	*	11/1986	Boebert et al.	707/8
5,204,812 A		4/1993	Kasiraj et al.	707/9
5,276,901 A		1/1994	Howell et al.	707/9
5,446,903 A	*	8/1995	Abraham et al.	710/240
5,499,371 A		3/1996	Henninger et al.	717/2
5,572,673 A	*	11/1996	Shuris	713/200
5,826,268 A		10/1998	Schaefer et al.	707/9
5,911,143 A	*	6/1999	Dienhart et al.	707/103
6,038,563 A	*	3/2000	Bapat et al.	707/10
6,105,027 A	*	8/2000	Schneider et al.	707/9
6,141,754 A	*	10/2000	Choy	713/200
6,292,798 B1	*	9/2001	Dockter et al.	707/9

OTHER PUBLICATIONS

Ravi S. Sandhu et al.: "Access Control: Principles and Practice" IEEE Communications Magazine, U.S. IEEE Service Center, Piscataway, N.J., vol. 32, No. 9, Sep. 1, 1994, pp. 40-48.

Jonathan Moffet et al.: "Specifying Discretionary Access Control Policy for Distributed Systems", Computer Communications, NL, Elsevier Science Publishers BV, Amsterdam, vol. 13, No. 9, Nov. 1, 1990, pp. 571-580.

Brian Moore, "Making a Secure Office System", ICL Technical Journal, GB, Peter Pergrinus Ltd. Hitchin, vol. 7, No. 4, Nov. 1, 1991, pp. 801-815.

* cited by examiner

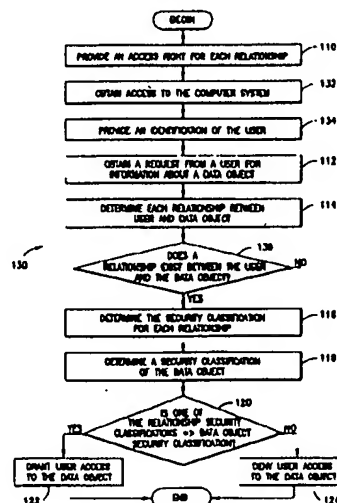
Primary Examiner—Greta L. Robinson

(74) Attorney, Agent, or Firm—Wiggin & Dana LLP; Gregory S. Rosenblatt; Michael K. Kinney

(57) **ABSTRACT**

A method for controlling access to information, which includes a plurality of data objects, on a computer system being accessible to a plurality of users is provided which generally comprises providing an access right for each relationship between a user and a data object, wherein each user can have a plurality of relationships to each data object, determining each relationship between the user and the data object when a user requests information about a data object, determining the security classification for each relationship between the user and the data object, and then granting the user access to the data object if one of the security classifications for all the relationships is equal to or greater than the security classification of the data object, and denying the user access to the data object if the security classifications for all the relationships is less than the security classification of the data object.

18 Claims, 10 Drawing Sheets



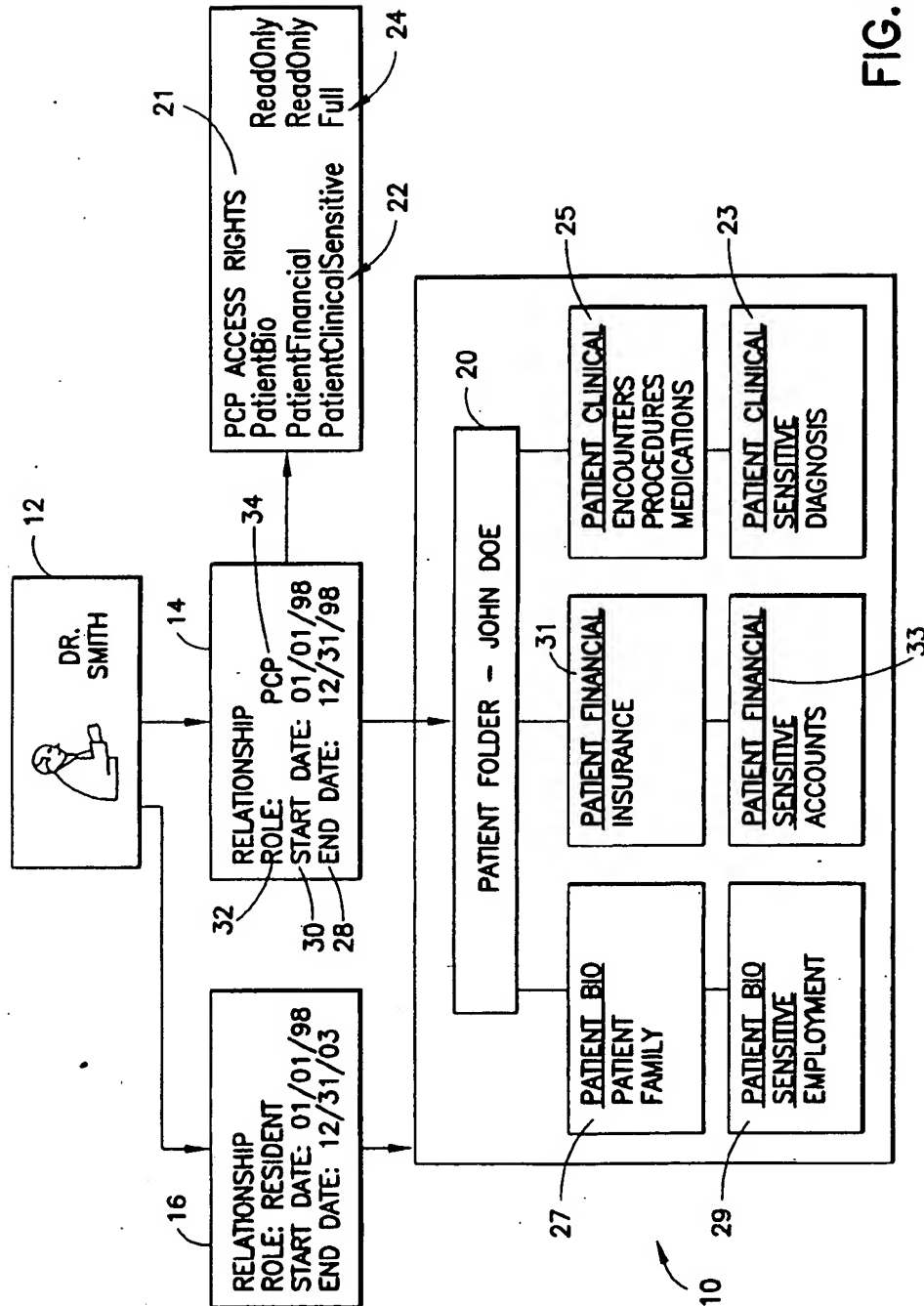


FIG. 1

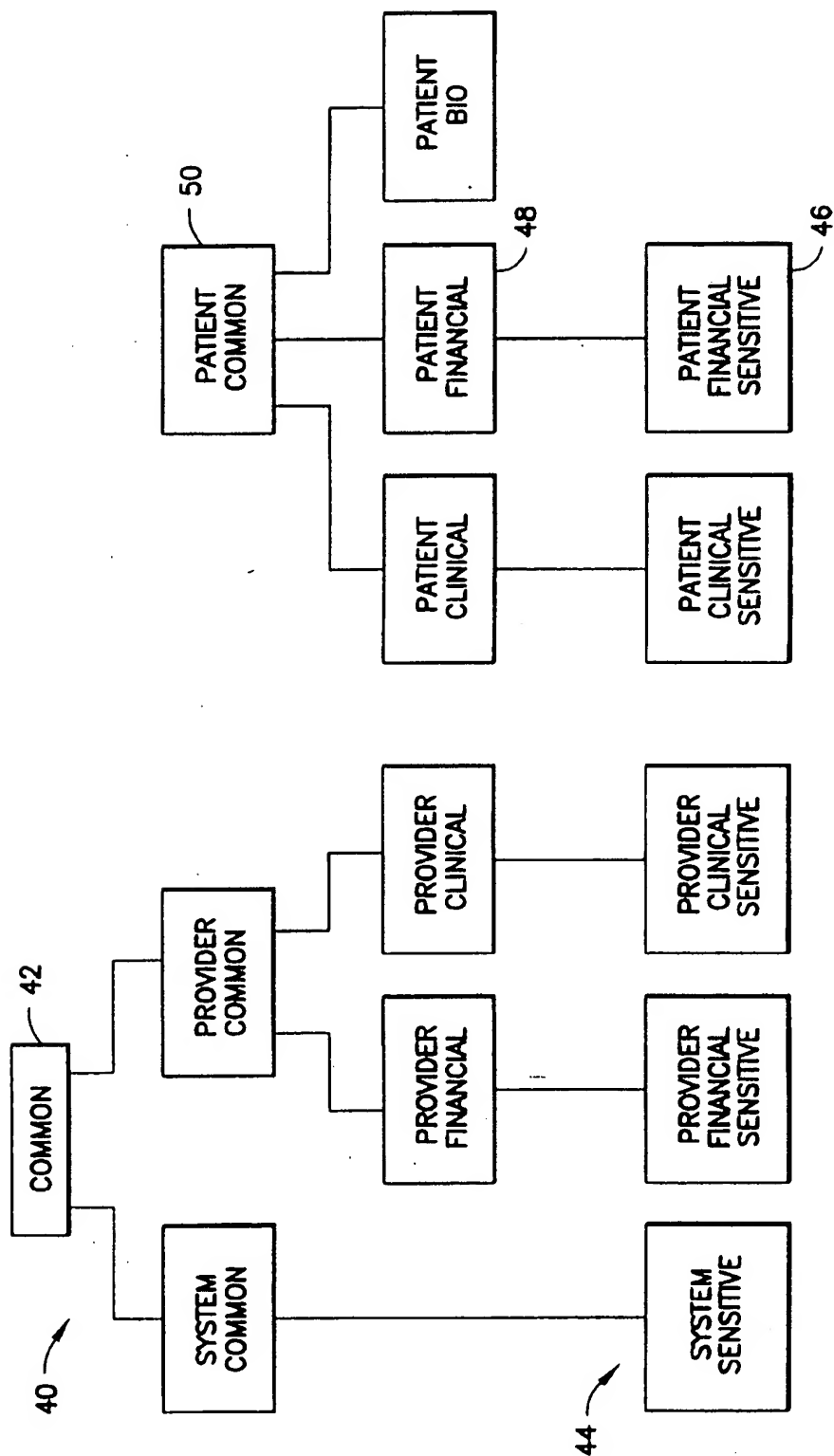


FIG. 2

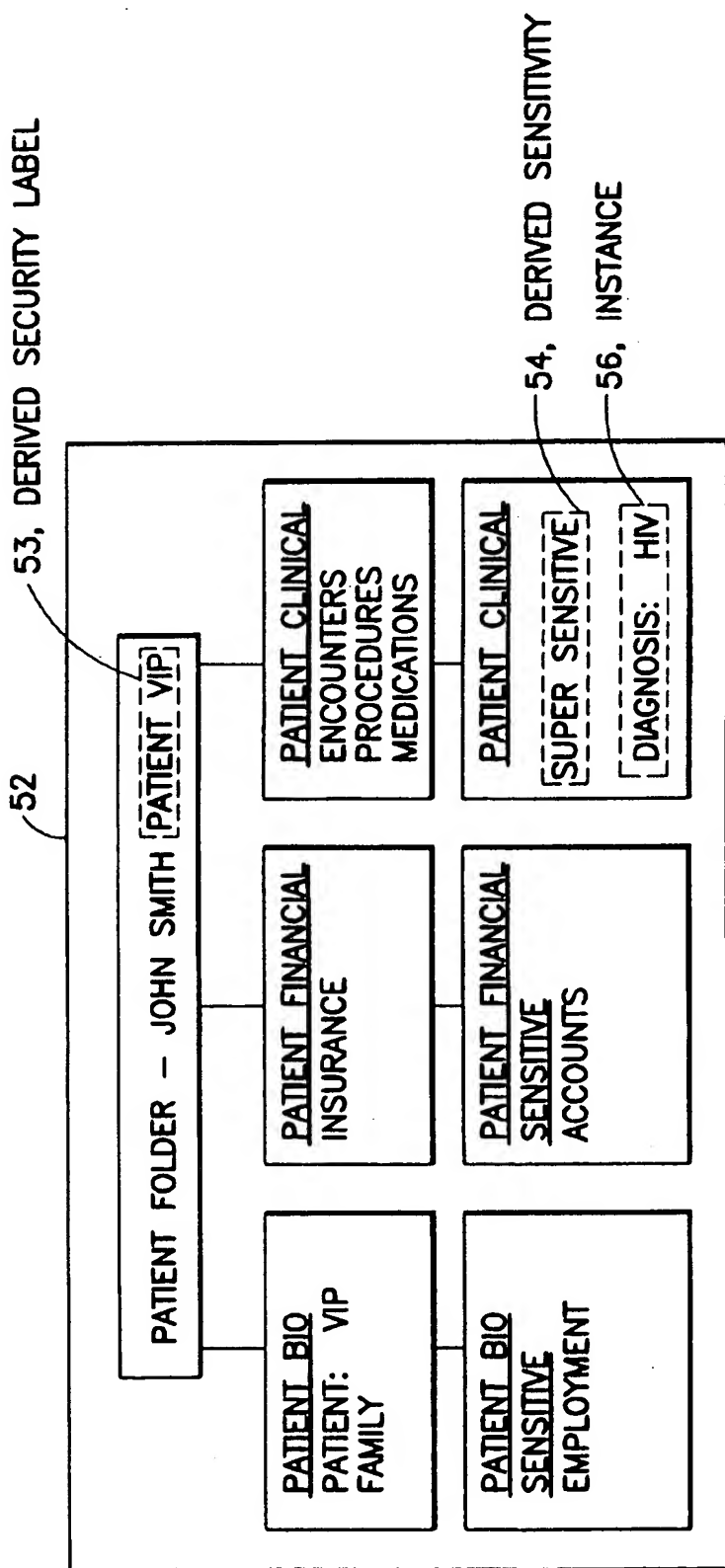


FIG. 3

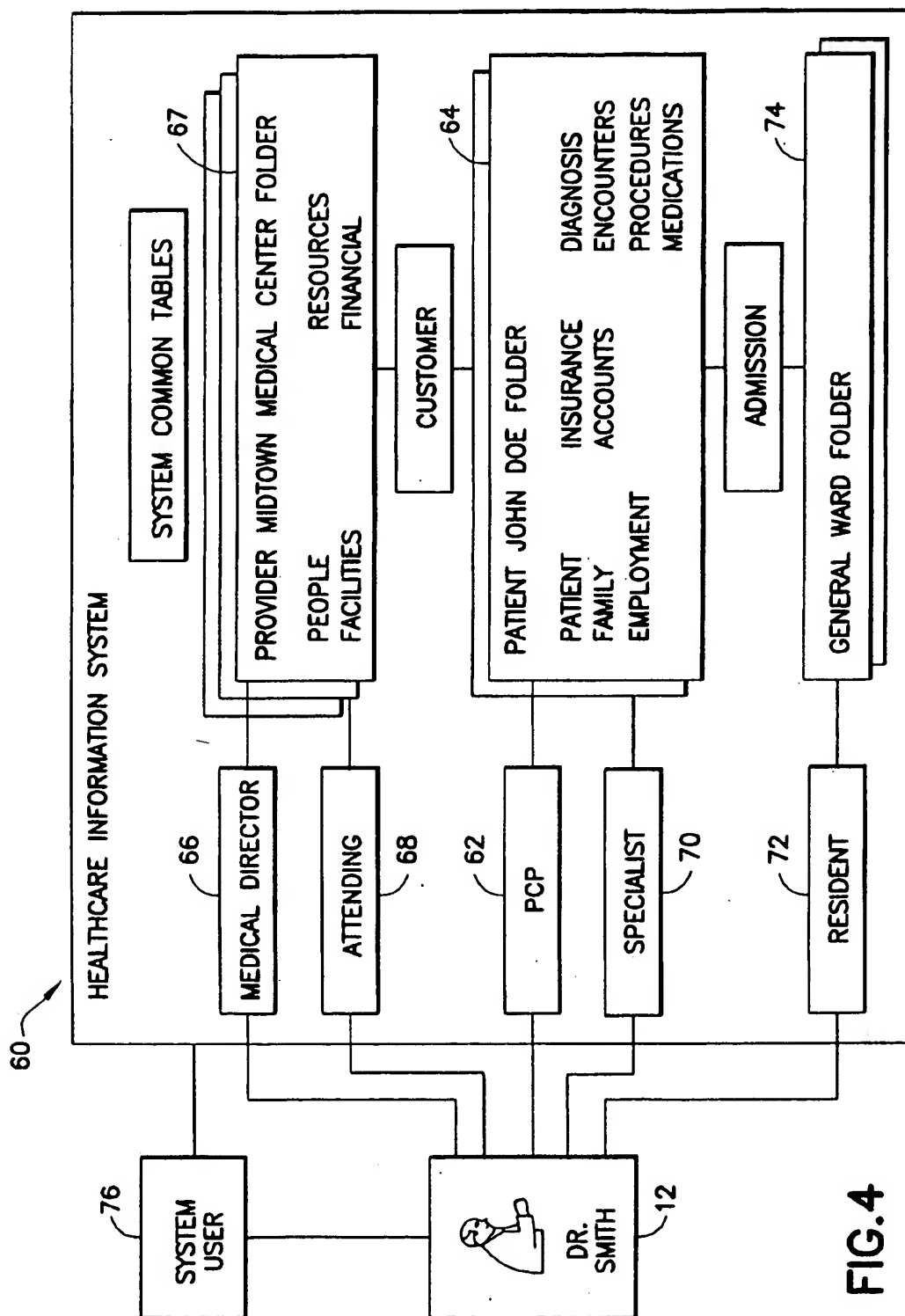
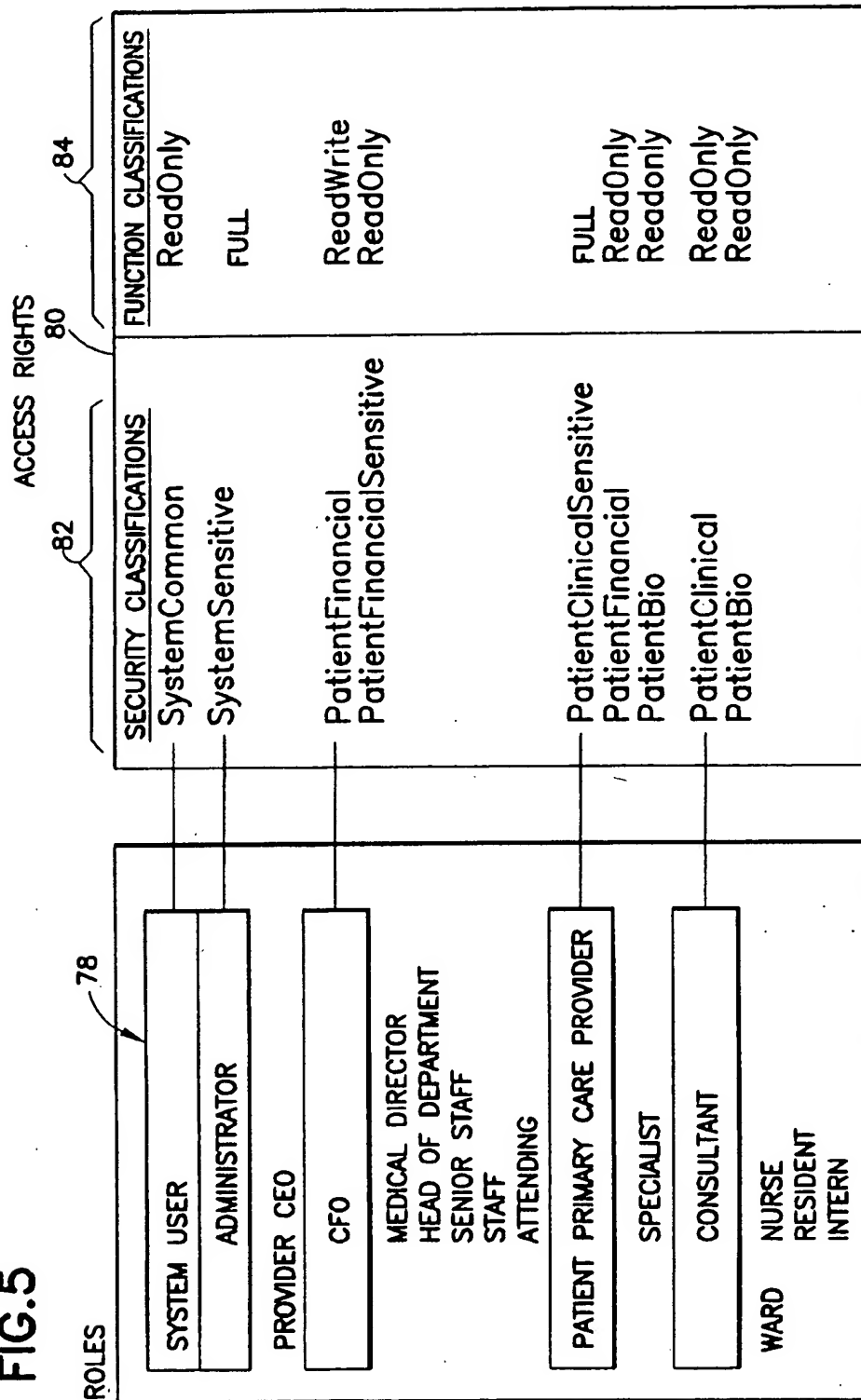


FIG. 4

FIG. 5



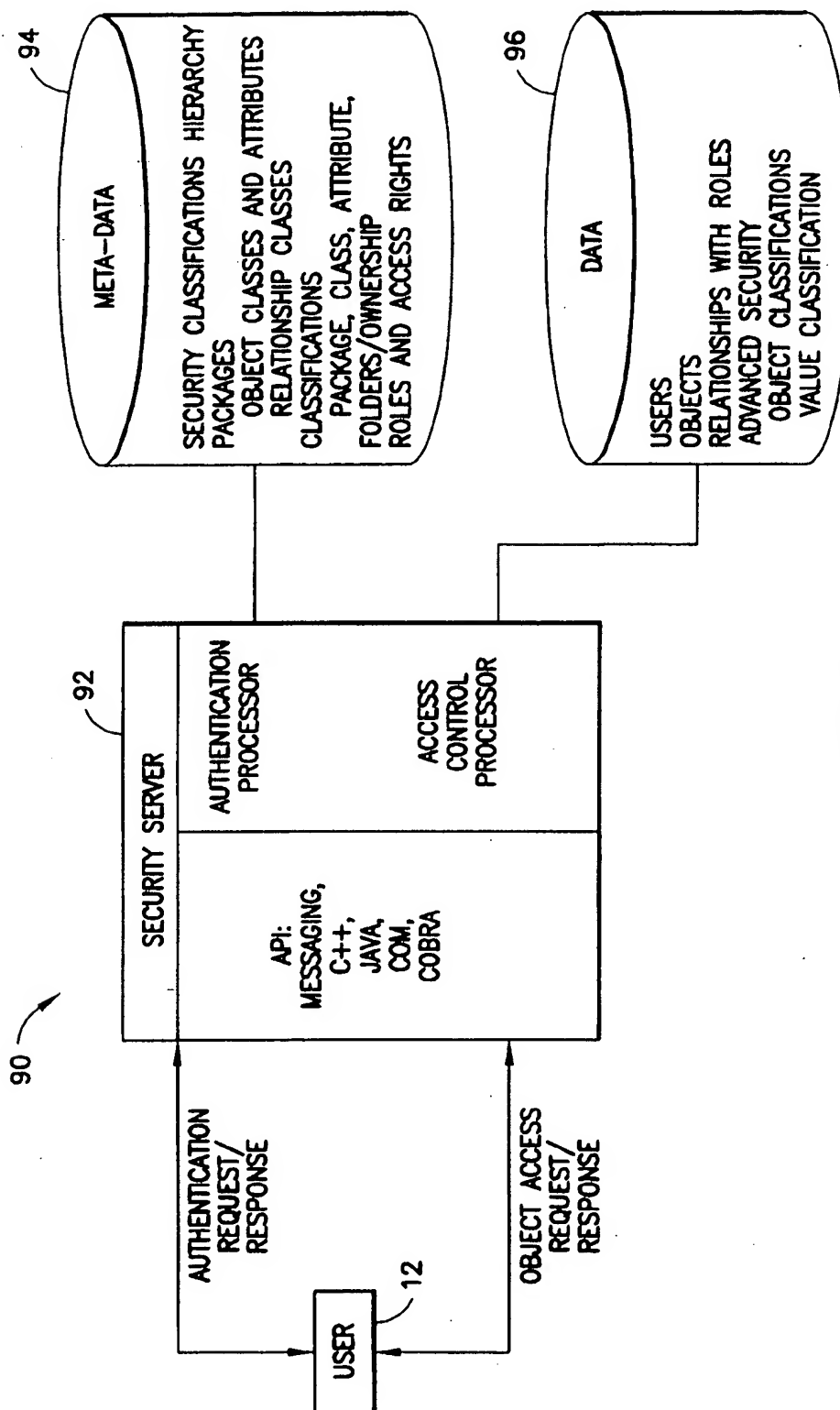


FIG. 6

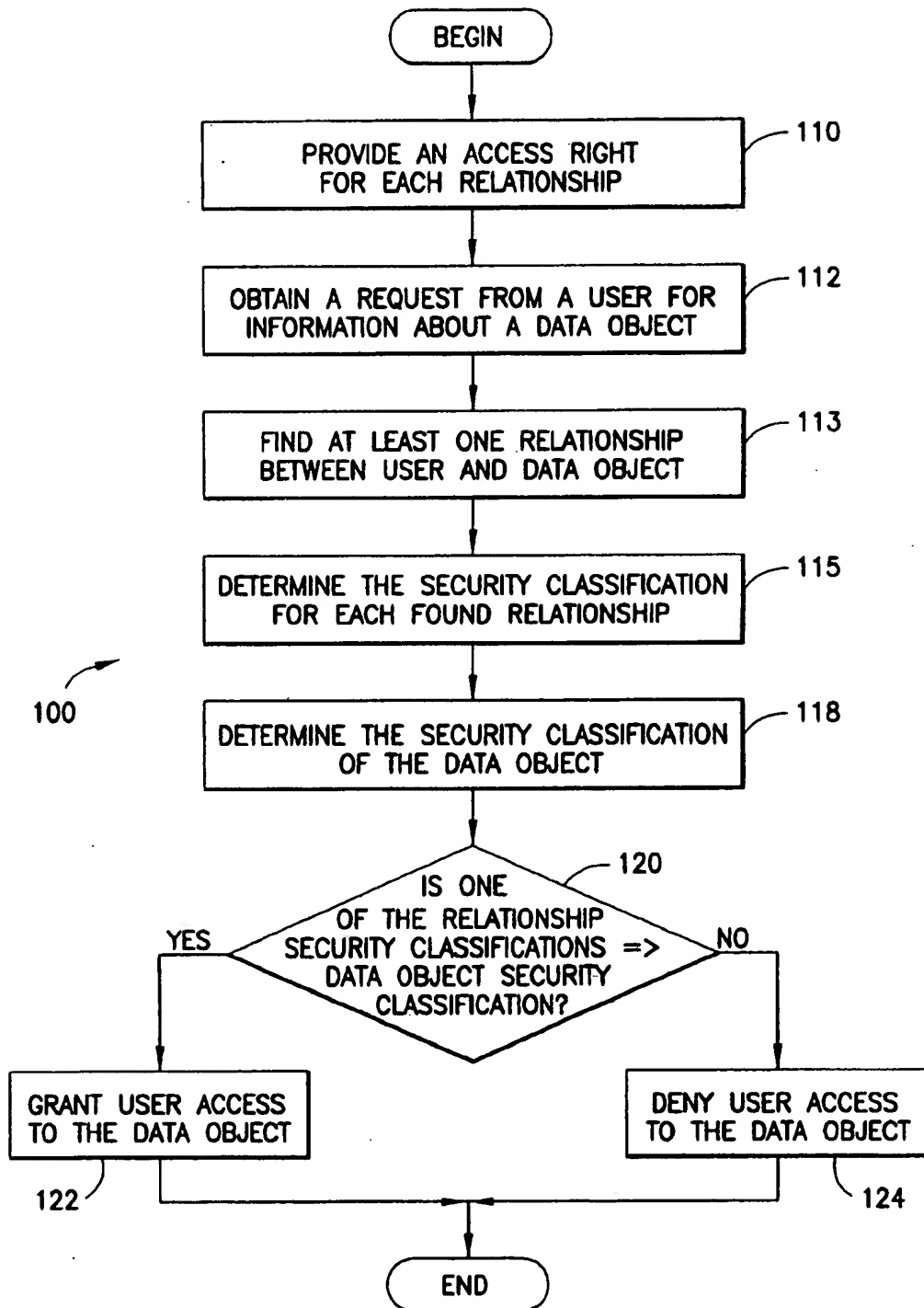
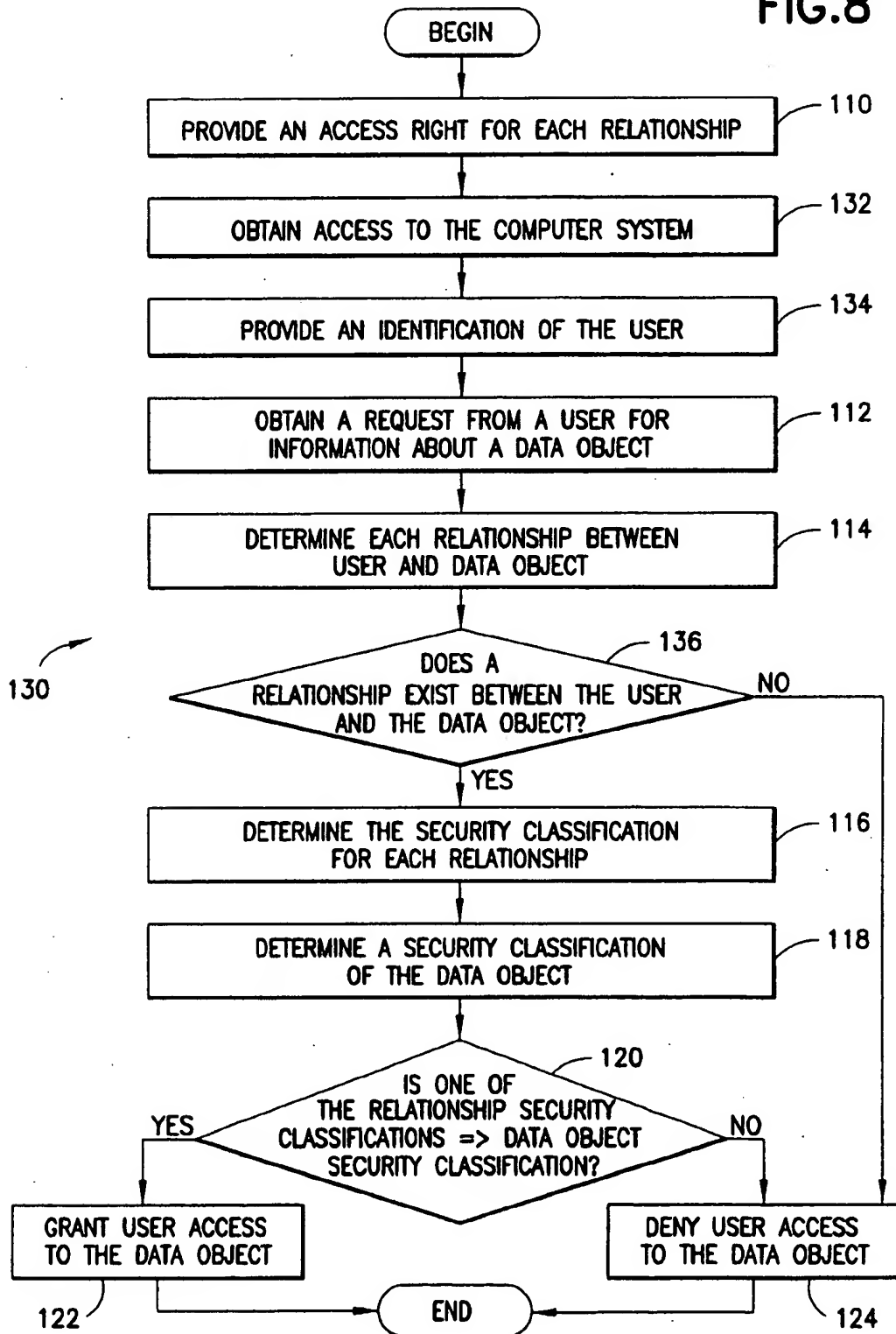
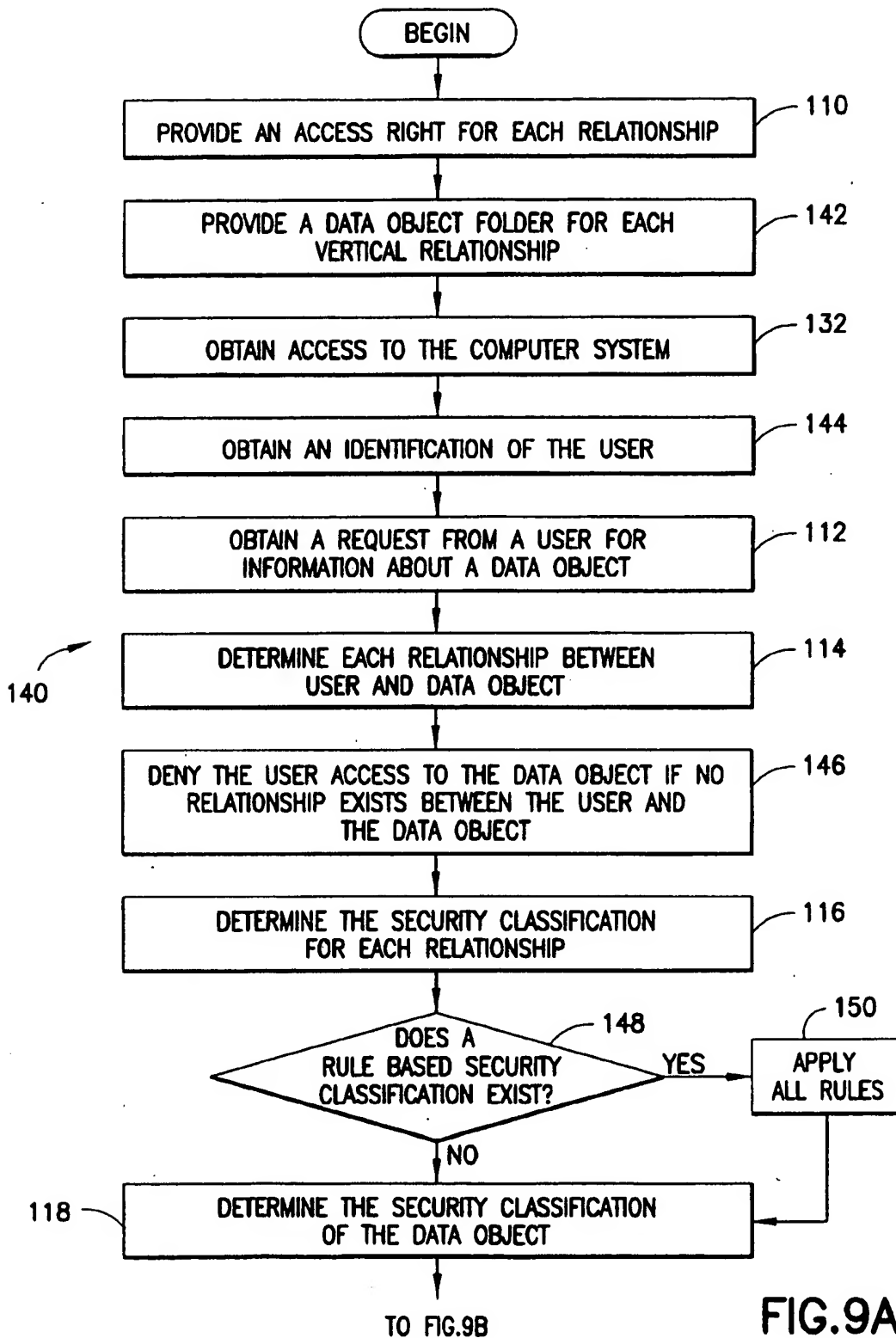
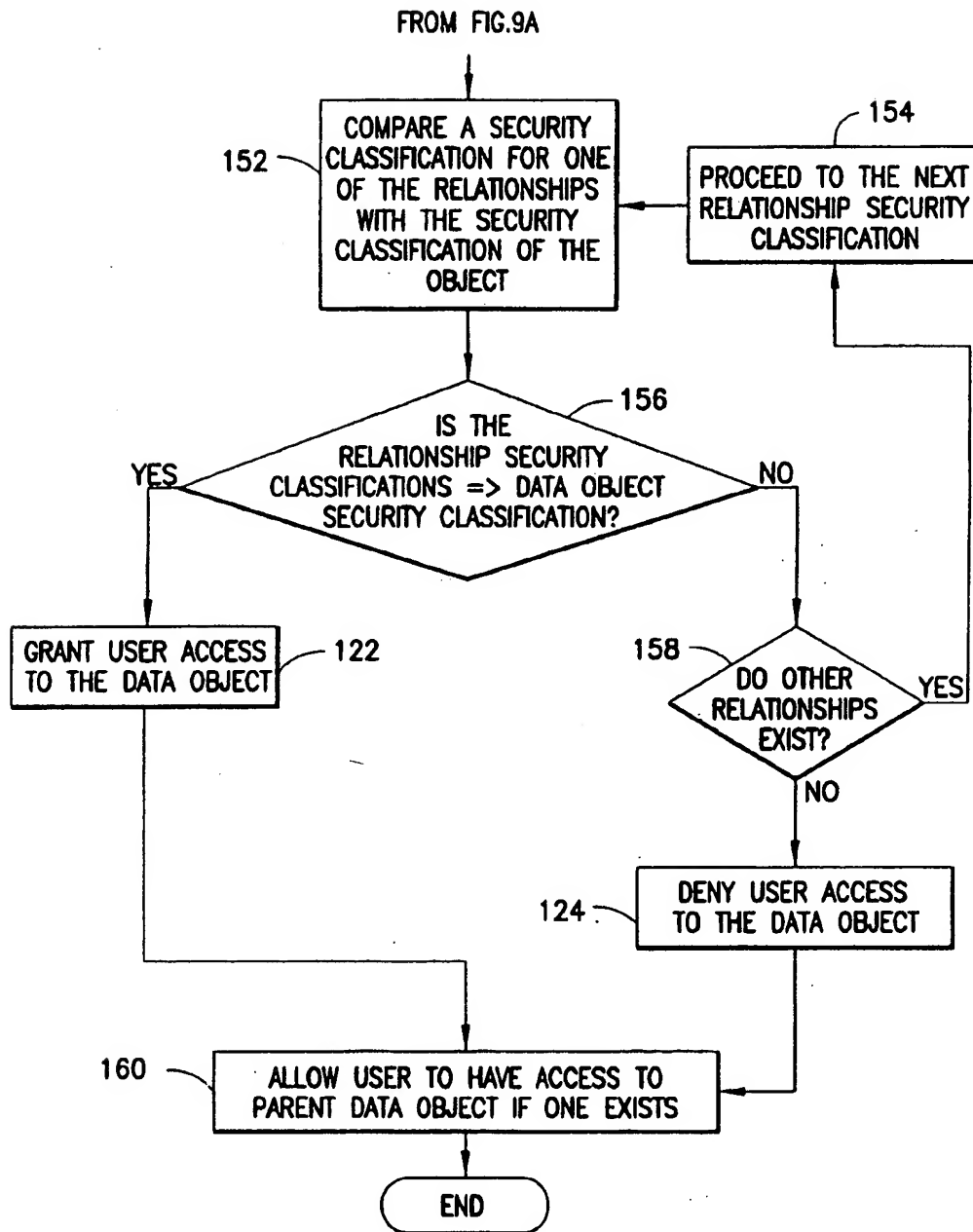


FIG.7

FIG. 8







1

METHOD FOR CONTROLLING ACCESS TO INFORMATION

RELATED APPLICATIONS

This application claims priority from U.S. Provisional Application No. 60/107,126 filed on Nov. 5, 1998, which is herein incorporated by reference.

FIELD OF THE INVENTION

The invention relates to a method of controlling access to information. More specifically, the invention relates to controlling user access to computer database information, possibly accessible in a client/server environment, in which the user may have an association with one or more objects in the database.

BACKGROUND OF THE INVENTION

As information has become more widely available to a larger number of corporate network users as well as to vendors, customers and the public, the need for precisely controlling access to this information has become paramount. Previous methods of access control, however, have not adequately met these needs.

Previous methods of access control focused on granting or denying access to classes of objects, but did not restrict access to specific objects, which is a practical necessity throughout business. Some examples of restricting access to specific objects include limiting patient information to only the caregivers that are currently treating them, limiting project information to current project team members, limiting department information to the current department employees, and limiting employee information to their current supervisors.

Even when an adequate level of access control has been achieved, it can be cumbersome to implement it because daily administration of access control lists may be required. For example, U.S. Pat. No. 5,276,901 to Howell et al. discloses a system for controlling group access to objects that uses group access control folders each having a security clearance. Folders can have a public access designation or an explicit access designation and/or a controlled access designation. For a user to have access to a folder having an explicit access designation, the user's ID must be listed explicitly within the folder. For a user to have access to a folder having a controlled access designation, the user must first have an affinity to the folder and can then access the folder if the user's clearance level is equal to or greater than the clearance level of the folder.

The disadvantage with the system of the '901 patent is that for a user who has an affinity to a folder having a controlled access designation and who is not listed in the folder's explicit access list, the system compares the clearance level of the folder to the clearance level of the user and not the clearance level of the affinity or relationship itself. Thus, this system thus does not provide for situations in which a user may have more than one relationship with a folder or an object wherein each relationship may have a different security level classification. Otherwise, to allow for flexibility of access in this system, the explicit access designation lists may have to be updated frequently, which can be time consuming.

U.S. Pat. No. 5,204,812 to Kasiraj et al. describes a method of controlling user access of documents based upon the relationship between the documents. Documents can be placed in a set comprising a linear relationship with the set

2

of documents as a whole having a sensitivity classification. User access is controlled by determining the classification of the user and comparing it the sensitivity classification of the set of documents.

U.S. Pat. No. 5,204,812 to Kasiraj et al. also describes a prior art document classification method in which documents are protected based upon their classification of use such as "loan application," while users are given classifications such as "loan officer." A system administrator would set up allowable document labels and retention periods such that, for instance, the loan officer could view the loan application for a period of three years while the loan is active. The methods and prior method described in the '812 patent to Kasiraj et al., however, also do not provide for multiple relationships between an object and the user.

What is desired, therefore, is an access control technology in which users can access only those data objects that they have a relationship or association with, wherein each user may have one or more relationships with a data object and each different relationship can have a different security classification. It is further desired that the system can control the types of functions a user can perform on the data object and that the system does not require daily administration of access control.

SUMMARY OF THE INVENTION

Accordingly, an object of the present invention is to provide a method for access control of information on a computer system in which users can access only those data objects to which they have an appropriate security classification.

It is a further object of the present invention to provide a method of the above type in which the system can precisely control the parts of the data object that the user can access.

In another embodiment of the present invention, it is an object to provide a method of the above type in which the system can precisely control the types of functions that can be performed on the data object once a user has access to the object.

It is still another object of the present invention to provide a method of the above type in which daily administration of the access control is unnecessary.

These objects of the invention are achieved by a method for controlling access to information, which includes a plurality of data objects, on a computer system being accessible to a plurality of users, wherein the method generally comprises providing an access right for each relationship between a user and a data object, wherein each user can have a plurality of relationships to each data object, determining each relationship between the user and the data object when a user requests information about a data object, determining the security classification for each relationship between the user and the data object, and then granting the user access to the data object if one of the security classifications for all the relationships is equal to or greater than the security classification of the data object, and denying the user access to the data object if the security classifications for all the relationships are less than the security classification of the data object.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of an example of file associated with an access control system in accordance with the present invention.

FIG. 2 is a block diagram of an example of security classifications for the present invention.

3

FIG. 3 is a block diagram of an example of rule driven security for the present invention.

FIG. 4 is a block diagram of an example of subject/object relationships for the present invention.

FIG. 5 is a block diagram showing examples of roles and associated access rights.

FIG. 6 is a block diagram of an example of security server in accordance with the present invention

FIG. 7 is a flow diagram showing the method of the present invention.

FIG. 8 is a more detailed flow diagram showing the method of the present invention.

FIGS. 9A and 9B are even more detailed flow diagrams showing the method of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

The present invention provides a method for controlling access to information on a computer system being accessible to a plurality of users. Each user of the system can have a plurality of relationships to each data object of information. When the user asks for information about the data object, the system determines each relationship between the user and the data object and determines the security classification for each relationship. Finally, the system then grants the user access to the data object if one of the security classifications for all the relationships is equal to or greater than the security classification of the data object, or denies the user access to the data object if all the security classifications are less than the security classification of the data object. The security classifications can also have a hierarchical structure.

The organization of information method of controlling access to information in accordance with the present invention is illustrated through the block diagram 10 shown in FIG. 1. A subject 12, Dr. Smith has a relationship with an object entitled John Doe's patient folder 20. By virtue of this relationship, Dr. Smith's has potential access to the patient folder 20 of John Doe. However, Dr. Smith does not have indiscriminate access to all parts of the folder. The relationship 14 between the subject 12 and the object 20 specifies the role 32 of the subject for the object. The role in this case is that of a Primary Care Provider (PCP) 34. The patient folder 20 is marked with security classification labels 22 that indicate the type of information from a security perspective and its sensitivity. Every role 34 is defined with access rights 21 that preferably include security classification labels 22 and function classifications 24. Thus, Dr. Smith has read-only access to the part of the folder that is marked as PatientBio 27, except the part that is marked PatientBio-Sensitive 29. He also has read-only access to the part of the folder that is marked as PatientFinancial 31, except the part that is marked PatientFinancialSensitive 33. He has full access to the parts of the folder marked PatientClinical 25, including the parts that are marked PatientClinicalSensitive 23.

Although this disclosure herein generally uses object-oriented terminology to describe the system, the access control system of the present invention is applicable to any type of information system, including object-oriented, relational and conceptual information systems. Table 1 lists the object-oriented terms used herein, and the equivalent terms currently used in relational and conceptual information systems.

4

TABLE 1

Object Oriented Terms, Equivalent Terms and Definitions.		
Term Used	Equivalent Terms	Definition
Object	Concept	A generic concept representing an abstract information item
Package	Database, Domain	A collection of objects, esp. classes and sub-packages
Class	Table, Type	A prototype for objects
Super Class		The class from which a given class is inherited
Sub Class		The classes that inherit from a given class
Attribute	Column, Property, Field	A property defined for a class
Instance	Row, Record, Tuple	A single object
Attribute Value	Cell, Value	A property value for an instance
Folder	Aggregate	An object and its sub-objects
Relationship Definition	Associations, Foreign Key	An association definition between classes
Relationship	Join, Link	An association between two instances

The system consists of objects at multiple levels of granularity, ranging from a set of databases to a single value in a record. Objects belong to Classes. For example, patient John Doe object belongs to the patient class. Relationship classes link object classes. For example, the patient-physician class links the physician class and the patient class.

Objects may have relationships with other objects. For example, patients can have relationships with physicians. Each relationship belongs to a relationship class. Each relationship is preferably a link between two objects.

A relationship class may define the formation of a folder. For example, as shown in FIG. 1, the relationship between a patient object 27 and a patient employment object 29 can be a vertical parent/child relationship that makes patient employment part of the patient folder. Thus, an object may consist of a number of sub-objects, forming a folder.

A patient's folder may consist of many other objects having subjects such as patient bio-data, insurance, financial, problems, encounters and treatment records.

An object at any level of granularity may be marked with a security classification that determines its accessibility and sensitivity level. For example, patient bio-data, family and employment records have a security classification label PatientBio 27. Optionally, the security classification of data objects may be derived using security rules. For example, the security classification of patient clinical records may be derived from the type of diagnosis performed.

Subjects are the users of the system who perform functions on objects. For example, physicians are a type of subject. Subjects can only gain access to objects by having relationships to objects. Each relationship provides access to the object's folder. For example, a physician may have a relationship 14 to a patient that defines the role 32 of the physician as the patient's "primary care provider," and thereby the physician has access to the patient's folder.

Each relationship 14 defines a role 34 that the subject is performing for the object. Each such role is defined with access rights 21 to items in the folder. Each access right specifies at least what security classification the object has and also can define what functions may be performed on objects of a given security classification in the folder. For example, a "Primary Care Provider" may have read-only

access to PatientBio information, but full access to Patient-Clinical information.

Optionally, subject/object relationships may also define context security rules that determine access based on the context of the relationship. For example, the physician-patient relationship may store the starting date 30 and ending date 28 of the relationship 14, which may be used to deny access before the starting date and after the ending date.

With the present invention, when a subject tries to perform a function on parts of an object's folder, the system examines the relationships between the subject and the target objects. It then examines the access rights 21 for the specified roles, and determines whether the subject has the rights to access the requested parts of the object's folder.

Object Model

Objects may be secured at many levels of granularity. At the most coarse level, the object to be secured may be the entire system, or a package. At the finest level, the object may be a single attribute value in a particular instance. Some of these objects are defined during system design and thus are part of the meta-data as shown in Table 2. Other objects are created during operation and are therefore part of the Operational data.

Following are the types of objects that may be secured:

TABLE 2

<u>Objects, Their Location in the System and Examples.</u>		
Secured Object	Location	Example
System		Contains all objects in the system
Package	Meta-data	All classes in the Clinical database.
Class	Meta-data	All instances in the Patient class
Folder	Meta-data	Patient John Doe's complete clinical information
		Provider Midtown Medical Center's complete business information
Attribute	Meta-data	Patient Diagnosis attribute for all patients in the Patient Class
Instance	Operational Data	Patient John Doe's instance
		Appointment for John Doe on June 5, 1989
Attribute Value	Operational Data	Patient John Doe's Diagnosis

Object classes are related to other object classes through relationship classes. These relationship classes are also object classes with their own attributes. For example, Table 3 shows the relationship classes and attributes that a health-care system may have:

TABLE 3

<u>Examples of Object Relationships Classes and their Attributes.</u>	
Relationship Class	Relationship Class Attributes
Patient - Physician	Physician Role, Start Date, End Date
Patient - Provider	Account Number, Start Date, Accounts Receivable
Provider - Physician	Role, ID, Start Date, End Date
Patient - Family Member	Relationship Type, Duration, Dependency
Patient - Treatment instances	Prescribed Medication
Floor - Physician	Role, Start DateTime, End DateTime
Patient - Floor	Room Number, Start DateTime, End DateTime

Relationship instances link object instances. For example, any number of instances of patient-physician relationship exist, linking physician instances to patient instances.

Object relationships play two key roles in the security system. Relationships between objects and their sub-objects create object folders. Relationships between subjects and objects form the basis of granting subjects access to their related objects.

Parent-child Object Relationships Establish Folders

Relationship classes define folders. For example, the patient's folder may consist of:

Family member relationships

Healthcare provider relationships and any owned instances of this relationship (e.g. charges and payments).

Patient encounter instances and any instances owned by the encounter instances.

Patient treatment instances and any instances owned by treatment instances.

Securing the master instance in a folder preferably secures all the owned folder instances. For example, if access to a patient's instance is denied, access to the patient's treatment instances is automatically denied without requiring any further security specification.

Folders may have subfolders. For example, a patient encounter may consist of dozens of sub-instances, which collectively form a folder. An operation performed during the encounter may itself be a subfolder. Securing a folder at any level secures all its subfolders.

The relationship class may define limitations on accessing subfolders. For example, the patient-provider relationship class may indicate read-only access to the patient sub-folder. Thus, users who have access to the provider's folder can only have read-only access to the patient's subfolders.

A logical system object forms a logical folder that owns all the objects in the system. The purpose of the system object is to allow access to objects that do not have any reason to have relationships to users. For example, generic tables such as information on drugs and diseases are accessible only through the system folder. Generally, only high level roles such as system administrator have full access to these objects, and most other system users have read-only access.

Object Security Classifications

Referring to FIG. 2, each object in a system 40 can be assigned a security classification label. The security classification label is a measure of the sensitivity of the information, and is the means through which access may be allowed or denied.

Security classification labels can be linked and can be formed in a hierarchical structure. A security classification label A may have a parent label B, so that granting access to A automatically grants access to B and all antecedents of B. The labels towards the root 42 of the hierarchy are for less sensitive information and have a lower security rank, whereas labels towards the leaf 44 are more sensitive and have a higher security rank. When a subject is provided access to a more sensitive label, they also have access to the less sensitive labels that are its antecedents. For example, referring to FIG. 2, if a user has access to Patient Financial Sensitive 46 information, they also have access to Patient Financial 48 and Patient Common 50 information. The purpose of linking 5 security classification labels is to simplify the task of assigning access rights.

There is no requirement in the system that all labels form a single hierarchy. In fact, the labels may form many

different kinds of patterns: single stand alone labels, linear ranked lists, or hierarchies. Any number of such patterns may exist in a single secured system 40.

Objects may be labeled with security classifications. Most of security classifications are on meta-data objects. More advanced security options may label operational data objects such as an instance or an attribute value.

If an object is not assigned a security classification label, it inherits its security classification label. The source of its inheritance is shown in Table 4:

TABLE 4

Inheritance of Security Classifications	
Secured Object Type	If no direct label, then the label is inherited from:
System	—
Package	Master Package (if any), Otherwise System
Class	Stricter of Package or Super Class
Folder	Owning Instance
Attribute	Class
Instance/Record	Class
Attribute Value	Stricter of Instance or Attribute

The system object preferably always has a default security classification label. If an object in the system does not have a direct security label and has no ancestors that have a security label, then it inherits the security label of the system object.

The security classifications for an instance or an attribute value may be obtained using derivation rules. For example, as shown with patient folder 52 in FIG. 3, the security classification for the patient diagnostic instance 56 may be derived from the sensitivity level of the problem diagnosed, thus allowing an extra level of security such as super sensitive 54 for patients diagnosed with certain types of diseases, such as HIV. Similarly, if an attribute in the patient's folder designates the patient as a VIP, the patient folder may be marked with a Patient VIP security label 53. A derived classification may be derived at the time the respective data is entered and stored as part of the instance, or it may be obtained at the time security is enforced. In other words, it may be persistent or non-persistent.

Subject-object Relationships and Roles

Subjects are a special class of objects that are the users of the system. Subject-object relationships form the basis of the access control of the present invention.

As shown in system 60 of FIG. 4, relationships between subjects and objects provide the basis for access control of the present invention. If a subject has a relationship to an object, then it has access to the folder of that object. The type of relationship is the subject's role for the object, wherein the role defines the precise access rights for the subject. For example, relationship between physician "Dr. Smith" 12 and patient "John Doe" may define the role of Dr. Smith as primary care physician (PCP) 62. Mr. Smith now has access to John Doe's folder 64, and has the access rights defined for the role "Primary Care Physician."

A subject may have any number of relationships. Each relationship has a unique role, which defines the precise rights that the subject has over the corresponding folders. For example, Mr. Smith may have a "Medical Director" role 66 at one provider having folder 67, an "Attending" role 68 at another provider, be the "Primary Care Physician" 62 for one patient, a "Specialist" role 70 for another patient, and act

as a "Resident" role 72 for the General ward having folder 74. Each of these relationships brings precise access to corresponding folders. FIG. 4 shows an example of these relationships.

Thus, a subject may have multiple relationships to the same object. For example, Dr. Smith may be the PCP for patient John Doe, a medical director for the provider who has the patient as a customer, and a resident for a ward where the patient is admitted. Each of these relationships defines a role for Dr. Smith and gives him specific rights to the Patient John Doe's folder. His full rights to this patient's folder is the aggregate of all such rights.

Although it is stated herein that the role defines the access rights, it should be understood that the relationship name itself can equally define the role and its access rights. For example, a doctor can have a "primary care provider" relationship and an "attending physician" relationship with a patient and these relationships can define the doctor's access rights to the patient's information.

Every subject preferably also has a relationship to the system, such as system user 76, which defines the basis for getting access to all the objects that the subject cannot access through any other relationship. This relationship is typically stored alongside the system's authentication information.

A subject is also stored as an object in the system, and preferably has a self role to its own object. Each type of subject may have a different self role, and each such role may define how the subjects may access or manipulate its own information. For example, a patient may have read-only access to his entire folder, but can modify only his bio-data information.

Roles and Access Rights

Referring to FIG. 5, each role 78 is defined with a specific set of access rights 80 to the target objects that are exposed through the relationship. A role 78 may have any number of access rights. Each access right 80 specification preferably consists of a security classification 82 and a function classification 84. The security classification 82 indicates the type of objects that may be accessed through this right. The function classification 84 indicates the type or types of functions that may be performed on the object.

Functions themselves can have a hierarchy, so that Full function includes the ReadWrite function, whereas the ReadWrite function includes the ReadOnly function. A generic set of such functions is defined by the system for all objects. In addition, each object may define its own set of functions.

In addition to roles, subject/object relationships preferably have other contextual information or rules that control the access to the object. Almost certainly, a relationship is likely to have the start and end time of the relationship. Subject/object relationships have additional security rules that control the access based on contextual information. For example, a rule that allows security access starting on the start date/time of relationship, and ending 24 hours after the ending date/time may be represented in computer code as:

```
Current_Date_Time BETWEEN Start_Date_Time AND (End_
Date_Time+ "24:00:00")
```

Implementing the Present Invention

The access control system may be implemented in a variety of ways. The options include:

integrated as part of a database management system to secure access to the database objects.

integrated in an object management system, such as an object request broker, to secure access to the objects accessed through the system.

implemented in a specific application such as a healthcare application.

implemented as a standalone security server. An example of a security server system 90 is shown in FIG. 6.

Security System Design

FIG. 6 shows an example of a security server. The steps for designing the security system design are:

1. Design a security classification hierarchy that serves the security needs of the organization.
2. Determine the necessary data 96. Design or reverse engineer the object classes and attributes of all the objects that need to be secured. Store these in meta-data 94.
3. Assure that object relationships are properly defined in the object model. Also mark the folder ownership to create logical folders. Store these in meta-data 94.
4. Mark all meta-data objects 94, i.e. packages, classes, attributes and functions, with appropriate security classifications. All unmarked objects will inherit their classifications.
5. Define a set of roles that encompass all the roles in all direct or indirect subject/object relationships in the system. Store these in meta-data 94.
6. Define a set of access rights for each role. Store these in meta-data 94.
7. Define context related security rules for each relationship. For example, if the relationship has a start and end date, then the rule may limit access only between those dates. Store these in meta-data 94.
8. Where required, define security rules to mark instance and attribute value security. Store these in operational data 96. This process defines all the meta-data requirements to enforce security. The system is now ready for operation.

Security Administration

Ongoing security administration needs with the present invention are comparatively limited. This administration can include when relationships are created between subjects and objects, such as assigning a physician to a patient or adding a user to the system, make sure that the role, start date/time, and end date/time are appropriately recorded. For the most part, this information alone drives the entire access enforcement process.

For more advanced security, individual instances may be marked with their own security classification. For example, a VIP patient may have a top-secret classification. Additionally, some attribute values may be stamped with their security classification. For example, instead of marking the whole VIP patient instance, only the diagnosis may be marked secret.

Flow Diagrams

Methods for controlling access to information on a computer system are shown in FIGS. 7-9B. Referring to FIG. 7, the system is first provided 110 as described above. The method of access control 130 is initiated when a subject tries to perform an operation on an object. The system will obtain 112 an access request from a user for information about a data object. The access request preferably contains: 1) the subject class and ID (at this point, the subject is already authenticated); 2) the instance class and ID; 3) the attribute or a list of attributes that are being accessed; and 4) the function to be performed.

The system will find 114 at least one of the relationships, if one exists, between the user and object. Alternatively, all the relationships that exist can be found. Generally, at least one relationship will be found, i.e. from the user to the system. If the user has other direct or indirect relationships to the object, additional relationships may be found. If only one of the relationships was initially found and access was not granted to the object, the system can continuously go back and find another relationship until it is conclusively determined that the user has access or does not.

After the relationships have been found between the user and the object, determine 116 the security classification from the access rights for each relationship. This step may comprise obtaining the role from each relationship, then obtaining the list of access rights for the role. Alternatively, the relationship itself can state the type of relationship. An access right may also comprise a functional classification limiting the functions that the user can perform on the object.

Next, determine the security classification labels for the object 118: For an instance security label, obtain the most restrictive of: the class label stored in the meta-data, the instance label stored in the operational data, or an instance label obtained from a security rule. For an attribute security label, obtain the most restrictive of: the attribute label class, the attribute label stored in the operational data, or an attribute label obtained from a security rule. If both an instance security label and an attribute security label exist, use the most restrictive of these.

Next, determine if the user can be granted access to the data object. If the security classification of the relationship is greater than the data object's security classification, then grant the user access 122 to the data object. Otherwise, if the security classification of all the relationships are less than the data object's security classification, then deny the user access 124 to the data object.

With the present invention, each relationship security classification can be compared with the security classification of the object until it is determined that the user has access. Alternatively, all relationship security classifications can be compared with the security classification of the object and the user access can be determined after the comparisons.

By stating that the security classification can be "greater," it should be understood that greater is defined as having a security level adequate to allow access to the data object. Additionally, other variations of methods can be used for determining the adequacy of the relationship's security level when compared to the data object's security level. For example, the system may require the user-object relationship to have a greater or equal security classification than the data object's security classification.

If the access to the data object is granted, it is also possible to add an entry to an available access right list noting the functions that are allowed to be performed by the access right's function classification. If the function the user desires to perform on the object that is in the list of available rights, then return "Access granted". Otherwise, return "Access denied" and optionally return the functions that are available. If the user's request consisted of multiple objects, for example multiple instances or attributes, the system can return an appropriate response for each object.

Referring to FIG. 8, a method for controlling access 130 can specifically include the steps of obtaining access 132 to the computer system and providing 134 and identification of the user. In this embodiment of the invention, the system determines each relationship 114 between the user and the

data object and determines the security classification for all relationships. Furthermore, the system can determine if no relationships exist 136 between the object and the user and deny access 124 to the data object on that basis.

Referring to FIGS. 9A and 9B, a method for controlling access 140 can also include rule based security classifications. The system will first determine if any security rules exist 148. If the instance has a non-persistent security label that is determined through one or more derivation rules, apply 150 the one or more rules to obtain the security labels. Depending on the needs of the system, the applicable security label can be the most or least restrictive of these security labels and the persistent security label. Preferably, however, the applicable security label is the most restrictive of these security labels.

If the invention described herein is made useable for a computer, it can be stored on a computer usable medium having computer readable program code means embodied therein for completing the method. The computer readable program code means can include any type of computer language available or a representation of the computer language thereof, such as machine code, assembly language, compiler language, alphanumeric code or binary code. The computer usable medium can include any magnetic, electric or optical device for storing computer language, such as hard disks, floppy disks, CD-ROMS, optical drives or zip drives.

It should be understood that the foregoing is illustrative and not limiting and that obvious modifications may be made by those skilled in the art without departing from the spirit of the invention. Accordingly, reference should be made primarily to the accompanying claims, rather than the foregoing specification, to determine the scope of the invention.

What is claimed is:

1. A method for controlling access to information on a computer system being accessible to a plurality of users, wherein the computer system has information including a plurality of data objects, the steps comprising:

providing an access right for each relationship between a user and a data object, wherein each user can have a plurality of relationships to each data object, wherein at least one user has a plurality of relationships with one data object, wherein each access right comprises a security classification, wherein some of the data objects have a vertical relationship defined by a parent data object and a child data object, and wherein for each vertical relationship the child data object has a more restrictive security classification than the parent data object;

obtaining a request from a user for information about a data object;

finding at least one of the relationships between the user and the data object;

determining the security classification for each relationship found between the user and the data object;

determining a security classification of the data object;

granting the user access to the data object if a level of one of the security classifications for all the relationships is greater than a level of the security classification of the data object;

granting the user access to a parent data object if the user has been granted access to a corresponding child data object; and

denying the user access to the data object if the security classifications for all the relationships are less than a level of the security classifications of the data object.

2. The method for controlling access to information according to claim 1, wherein each access right further comprises a function classification for specifying one or more functions that may be performed on the data object, further comprising the step of limiting functional access to the data object to the one or more functions specified by the function classification.

3. The method for controlling access to information according to claim 1, further comprising the step of applying at least one rule to determine the security classification between the user and the data object.

4. The method for controlling access to information according to claim 1, wherein the computer system is a server and the user is a client of the server, and further comprising the step of allowing a user to obtain access to the server.

5. The method for controlling access to information according to claim 1, wherein the computer system is a local area network server and the user is a client of the server, and further comprising the step of allowing a user to obtain access to the local area network server.

6. The method for controlling access to information according to claim 1, further comprising the step of obtaining access to the data object after being granted access to the data object.

7. The method for controlling access to information according to claim 1, further comprising the step of denying the user access to the data object if no relationship exists between the user and the data object.

8. A method for controlling access to information on a computer system being accessible to a plurality of users, wherein the computer system has information including a plurality of data objects, the steps comprising:

providing an access right for each relationship between a user and a data object, wherein each user can have a plurality of relationships to each data object, wherein at least one user has a plurality of relationships with one data object, wherein each access right comprises a security classification, wherein some of the data objects have a vertical relationship defined by a parent data object and a child data object, and wherein for each vertical relationship the child data object has a more restrictive security classification than the parent data object;

creating a data object folder for each vertical relationship; obtaining a request from a user for information about a data object;

finding at least one of the relationships between the user and the data object;

determining the security classification for each relationship found between the user and the data object;

determining a security classification of the data object;

granting the user access to the data object if a level of one of the security classifications for all the relationships is greater than a level of the security classification of the data object;

granting the user access to a parent data object if the user has been granted access to a corresponding child data object; and

denying the user access to the data object if the security classifications for all the relationships are less than a level of the security classifications of the data object.

9. A method for controlling access to information on a computer system being accessible to a plurality of users, wherein the computer system has information including a plurality of data objects, the steps comprising:

13

providing an access right for each relationship between a user and a data object, wherein each user can have a plurality of relationships to each data object, wherein at least one user has a plurality of relationships with one data object, and wherein each access right comprises a security classification and a function classification for specifying one or more functions that may be performed on the data object, wherein the security classifications are arranged in a hierarchical structure and wherein some of the data objects can have a vertical relationship defined by a parent data object and a child data object, wherein for each vertical relationship the child data object has a more restrictive security classification than the parent data object;

obtaining access to the computer system;

providing an identification of a user;

providing a request from the user for information about a data object;

determining a security classification of the data object;

granting the user access to the data object and limiting functional access to the data object to the one or more functions specified by the function classification if a level of one of the security classifications for all the relationships is equal to or greater than a level of the security classification of the data object;

providing a data object folder for each vertical relationship;

granting the user access to a parent data object if the user has been granted access to a corresponding child data object;

denying the user access to the data object if no relationship exists between the user and the data object; and

denying the user access to the data object if the level of the security classification for all the relationships are less than the level of the security classification of the data object.

10. The method for controlling access to information according to claim 9, further comprising the steps of:

determining each relationship between the user and the data object; and

comparing the security classification for each relationship with the security classification of the data object.

11. The method for controlling access to information according to claim 9, further comprising the step of applying at least one rule to determine the security classification for the relationship between the user and the data object.

12. The method for controlling access to information according to claim 9, wherein the computer system is a server and the user is a client of the server, and wherein the step of obtaining access to the computer system comprises obtaining access to the server.

13. The method for controlling access to information according to claim 9, wherein the computer system is a local area network server and the user is a client of the server, and wherein the step of obtaining access to the computer system comprises obtaining access to the local area network server.

14. A method for controlling access to information on a computer system being accessible to a plurality of users, wherein the computer system has information including a plurality of data objects, wherein the security classifications are arranged in a hierarchical structure, wherein some of the data objects can have a vertical relationship defined by a parent data object and a child data object, wherein for each vertical relationship the child data object has a more restrictive security classification than the parent data object, the steps comprising:

14

providing an access right for each relationship between a user and a data object, wherein each user can have a plurality of relationships to each data object, wherein at least one user has a plurality of relationships with one data object, and wherein each access right comprises a security classification and a function classification for specifying one or more functions that may be performed on the data object;

providing a data object folder for each vertical relationship;

allowing access to the computer system;

obtaining an identification of a user;

obtaining a request from the user for information about a data object;

determining each relationship between the user and the data object;

denying the user access to the data object if no relationship exists between the user and the data object;

comparing the security classification for each relationship with the security classification of the data object;

granting the user access to the data object and limiting functional access to the data object to the one or more functions specified by the function classification if a level of one of the security classifications for all the relationships is equal to or greater than a level of the security classification of the data object;

granting the user access to a parent data object if the user has been granted access to a corresponding child data object; and

denying the user access to the data object if the level of the security classification for all the relationships are less than the level of the security classification of the data object.

15. An article of manufacture, comprising:

a computer usable medium having computer readable program code means embodied therein for controlling access to information on a computer system being accessible to a plurality of users, wherein the computer system has information including a plurality of data objects wherein some of the data objects can have a vertical relationship defined by a parent data object and a child data object, wherein the child data object has a more restrictive security classification than the parent data object, the steps comprising:

computer readable program code means for causing the computer system to provide an access right for each relationship between a user and a data object, wherein each user can have one or more relationships to each data object, and wherein each access right comprises a security classification;

computer readable program code means for causing the computer system to provide a data object folder for each vertical relationship;

computer readable program code means for causing the computer system to grant the user access to a parent data object if the user has access to the child data object;

computer readable program code means for causing the computer system to accept a request from a user for information about an object;

computer readable program code means for causing the computer system to determine each relationship between the user and the data object;

computer readable program code means for causing the computer system to determine the security classification for each relationship between the user and the data object;

15

computer readable program code means for causing the computer system to determine a security classification of the data object;

computer readable program code means for causing the computer system to grant the user access to the data object if a level of one of the security classifications for all the relationships is equal to or greater than a level of the security classification of the data object; and

computer readable program code means for causing the computer system to deny the user access to the data object if the level of the security classification for all the relationships are less than the level of the security classification of the data object.

16. The article of manufacture according to claim 15, further comprising computer readable program code means for causing the computer system to compare the security

16

classification for each relationship with the security classification of the data object.

17. The article of manufacture according to claim 15, wherein each access right further comprises a function classification for specifying one or more functions that may be performed on the data object, further comprising computer readable program code means for causing the computer system to limit functional access to the data object to the one or more functions specified by the function classification.

18. The article of manufacture according to claim 15, wherein the security classifications are further determined by rules, and further comprising computer readable program code means for causing the computer system to apply at least one rule to determine an additional security classification for each relationship between the user and the data object.

* * * * *